Technical
Instructions

IDC DOCUMENTATION

Threshold Monitoring Subsystem Software User Manual

Notice

This document was published April 2001 by the Monitoring Systems Operation of Science Applications International Corporation (SAIC) as part of the International Data Centre (IDC) Documentation. Every effort was made to ensure that the information in this document was accurate at the time of publication. However, information is subject to change.

Contributors

Tormod Kværna, NORSAR
Frode Ringdal, NORSAR
Lyla Taylor, NORSAR

Ordering Information

The ordering number for this document is SAIC-01/3004.

This document is cited within other IDC documents as [IDC6.5.14].

Threshold Monitoring Subsystem Software User Manual

CONTENTS

About this Document i

- PURPOSE ii
- SCOPE ii
- AUDIENCE ii
- RELATED INFORMATION ii
- USING THIS DOCUMENT iii
  - Conventions iv

Chapter 1: Introduction 1

- SOFTWARE OVERVIEW 2
- STATUS OF DEVELOPMENT 5
- FUNCTIONALITY 5
  - Features and Capabilities 9
  - Performance Characteristics 14
  - Related Tools 14
- INVENTORY 15
  - Software 15
  - Parameters 18
- ENVIRONMENT AND STATES OF OPERATION 21
  - Software Environment 21
  - Normal Operational State 21
  - Contingencies/Alternate States of Operation 23

Chapter 2: Operational Procedures 25

- SOFTWARE STARTUP 26
- SOFTWARE SHUTDOWN 28
- BASIC PROCEDURES 28

- ADVANCED PROCEDURES 28
  - Creating TM Sessions 29
  - Adding Stations to Existing Session 29
  - Deleting Stations from Existing Session 32
  - Copying Data from Disk Loops 33
- MAINTENANCE 36
  - Tuning Processing Parameters 36

Chapter 3: Troubleshooting 49

- MONITORING 50
- INTERPRETING ERROR MESSAGES 50
  - General Advice 50
  - TMthreshold 55
  - TMmap 60
  - TMprod and TMpostreb 63
  - CreateTMSession, AddTMStation, and DeleteTMStation 63
- SOLVING COMMON PROBLEMS 72
- REPORTING PROBLEMS 73

Chapter 4: Installation Procedures 75

- PREPARATION 76
  - Obtaining Released Software 76
  - Hardware Mapping 76
- EXECUTABLE FILES 76
- CONFIGURATION DATA FILES 77
  - Environment Variables 77
  - DFX 79
  - TMthreshold 85
  - TMmap 86
  - Static Data 87
- DATABASE 98
  - Accounts 98
  - Tables 98

- TUXEDO FILES 99
- INITIATING OPERATIONS 103
  - CreateTMSession Input 104
  - CreateTMSession Output 106
- VALIDATING INSTALLATION 112

References 113

Appendix: PIDC Tuning Results A1

Glossary G1

Index I1

Threshold Monitoring Subsystem Software User Manual

FIGURES

FIGURE 1.	IDC SOFTWARE CONFIGURATION HIERARCHY	3
FIGURE 2.	TM PROCESSING FLOW	4
FIGURE 3.	TM PROCESSING HIERARCHY	7
FIGURE 4.	TM DISPLAY OF DATA AVAILABILITY AND INTERFERING EVENTS	11
FIGURE 5.	TM DISPLAY OF BACKGROUND NOISE LEVEL AND SIGNALS	12
FIGURE 6.	TM DISPLAY OF AVERAGE AND POOREST NETWORK DETECTION CAPABILITY	13
FIGURE 7.	STATIC DATA DIRECTORY	20
FIGURE 8.	TM DATA FLOW	22
FIGURE 9.	AVERAGE NARROWBAND LOG(STA) OF ILAR NOISE SEGMENTS	38
FIGURE 10.	SNR (STA/LTA) OF ILAR EVENTS VERSUS EVENT DISTANCE	39
FIGURE 11.	ATCOMP DATA FOR DISTANCES > 15 DEG.	41
FIGURE 12.	ATCOMP DATA FOR DISTANCES < 15 DEG.	42
FIGURE 13.	BEAMFORMING STEERING POINTS FOR ASSESSING BEAM MIS-STEERING SIGNAL LOSS	44
FIGURE 14.	ILAR SIGNAL LOSS FOR EVENTS FURTHER THAN 15 DEG.	45
FIGURE 15.	ILAR SIGNAL LOSS FOR EVENTS WITHIN 15 DEG.	46
FIGURE 16.	ILAR BEAM DEPLOYMENT FOR 15–180 DEG.	47
FIGURE 17.	ILAR BEAM DEPLOYMENT FOR 1.4–15 DEG.	48
FIGURE 18.	GLOBAL TARGET POINTS	97
FIGURE 19.	SESSION DIRECTORY STRUCTURE	107

Threshold Monitoring Subsystem Software User Manual

TABLES

Table I:	Data Flow Symbols	iv
Table II:	Typographical Conventions	v
Table III:	Technical Terms	v
Table 1:	LoopCopy Parameters	34
Table 2:	Disk Loop Writing Codes	52
Table 3:	DFX Detection Parameters	80
Table 4:	DFX TM Recipe Parameters	81
Table 5:	Beam Definition Parameters	82
Table 6:	Beam Weight Parameters	84
Table 7:	TMthreshold Parameters	85
Table 8:	TMmap Parameters	87
Table 9:	beambasic Parameters	88
Table 10:	Missteer.Dlow-Dhig Parameters	89
Table 11:	TM.site Parameters	90
Table 12:	tab.PTM Parameters	92
Table 13:	tmstatus.dat Parameters	93
Table 14:	data.cpt Parameter Files	94
Table 15:	cur_pri.data Parameters	95
Table 16:	offset.data Parameters	96
Table 17:	targets.n Parameters	97
Table 18:	TM fpdescription Rows	98
Table 19:	tuxshell Parameters	99
Table 20:	CreateTMSession Parameters	104
Table A-1:	Array Configurations	A1
Table A-2:	Array Tuning Parameters	A3

[TABLE A-3: ARRAY BEAM STEERING PARAMETERS A4

TABLE A-4: 3-C STATION TUNING PARAMETERS A9

About this Document

This chapter describes the organization and content of the document and includes the following topics:

- Purpose
- Scope
- Audience
- Related Information
- Using this Document

About this Document

PURPOSE

This document describes how to use the Threshold Monitoring (TM) Subsystem software of the International Data Centre (IDC). The software is part of the Performance Monitoring computer software component (CSC) of the System Monitoring Computer Software Configuration Item (CSCI) and is identified as follows:

Title:	Threshold Monitoring Subsystem
Abbreviation:	TM

SCOPE

This manual describes TM including its overall processing flow and the tuning procedure for including new stations in the subsystem. With this manual at hand, you should be able to install, run, and maintain TM at the IDC. This document does not describe the software's design or requirements. These topics are described in sources cited in "Related Information."

AUDIENCE

This document is intended for the operator of the software. However, less experienced users may find certain sections useful as a description of the subsystem.

RELATED INFORMATION

The following documents complement this document:

- *Database Schema, Revision 2* [IDC5.1.1Rev2]

- *Configuration of PIDC Databases* [IDC5.1.3Rev0.1]
- *Distributed Application Control System (DACS) Software User Manual* [IDC6.5.2Rev0.1]
- *Threshold Monitoring Subsystem* [IDC7.1.14] (software design document)

See "References" on page 113 for a list of documents that supplement this document.

USING THIS DOCUMENT

This document is part of the overall documentation architecture for the IDC. It is part of the Technical Instructions category, which provides guidance for installing, operating, and maintaining the IDC systems. This document is organized as follows:

- Chapter 1: Introduction

  This chapter provides an overview of the software's capabilities, development, and operating environment.

- Chapter 2: Operational Procedures

  This chapter describes how to use the software and includes detailed procedures for startup and shutdown, basic and advanced features, security, and maintenance.

- Chapter 3: Troubleshooting

  This chapter describes how to identify and correct common problems related to the software.

- Chapter 4: Installation Procedures

  This chapter describes first how to prepare for installing the software, then how to install the executable files, configuration data files, database elements, and Tuxedo files. It also describes how to initiate operation and how to validate the installation.

- References

  This section lists the sources cited in this document.

- Appendix: PIDC Tuning Results

  This appendix lists the tuned TM parameters at the time of publication.

- Glossary

  This section defines the terms, abbreviations, and acronyms used in this document.

- Index

  This section lists topics and features provided in this document along with page numbers for reference.

Conventions

This document uses a variety of conventions, which are described in the following tables. Table I shows the conventions for data flow diagrams. Table II lists typographical conventions. Table III explains certain technical terms that are not part of the standard Glossary, which is located at the end of this document.

TABLE I: DATA FLOW SYMBOLS

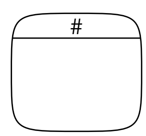
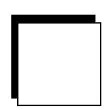
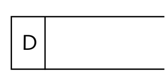
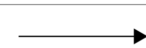
Description	Symbol
process	
external source or sink of data	
D = disk store Db = database store	
control flow	
data flow	

TABLE II: TYPOGRAPHICAL CONVENTIONS

Element	Font	Example
database table	**bold**	**fpdescription**
database table and attribute, when written in the dot notation		**fpdescription.**_lddate_
database attributes	_italics_	_lddate_
processes, software units, and libraries		_TMthreshold_
user-defined arguments and variables used in parameter (par) files or program command lines		`CreateTMSession par=`<_parfile_>
titles of documents		_Distributed Application Control System (DACS) Software User Manual_
computer code and output	`courier`	`>(list ‘a ‘b ‘c)`
filenames, directories, and websites		`bin/`
text that should be typed exactly as shown		`role=TMprod`
`%` is used as the command-line prompt		`% rlogin` <_machine_>

TABLE III: TECHNICAL TERMS

Term	Description
invoke	Execute a program.
LEBDB	Late Event Bulletin database. Database from which analysts work.

Chapter 1: Introduction

This chapter provides a general description of the software and includes the following topics:

- Software Overview
- Status of Development
- Functionality
- Inventory
- Environment and States of Operation

Chapter 1: Introduction

SOFTWARE OVERVIEW

Figure 1 shows the logical organization of the IDC software. TM is part of the Performance Monitoring CSC of the System Monitoring CSCI. Figure 2 shows the relationship of TM to the Station Processing CSC and the Interactive Processing, Distributed Processing, and Data Services CSCIs.

Data for TM calculations are the continuous seismic data from the stations of the primary seismic network, which are recorded on disk loops (D in Figure 2) by the Data Services CSCI. The seismic data for each station are checked for quality, beamformed (arrays only), bandpass filtered, and short-term-averaged (STA) using the Detection and Feature Extraction software (*DFX*) of the Station Processing CSC [Wah96]. The *DFX* program is run in the station processing pipeline, processing 10-minute data segments as soon as complete segments are recorded and available at the IDC. The continuous STA data are then stored on new disk loops with a typical sampling interval of one second. Data gaps and processing gaps are identified by NULL values in the disk loops.

Network detection thresholds are calculated by *TMthreshold* (process 1 in Figure 2). This program reads the STA disk loops for each of the primary seismic stations, calculates the 90 percent detection thresholds for each of 2,562 globally distributed target areas at 10-second time intervals, and stores the results on disk loops. *TMthreshold* is run in the Standard Event List 3 (SEL3) pipeline, which initiates processing of 20-minute time segments with a time delay of 10 hours behind real time.

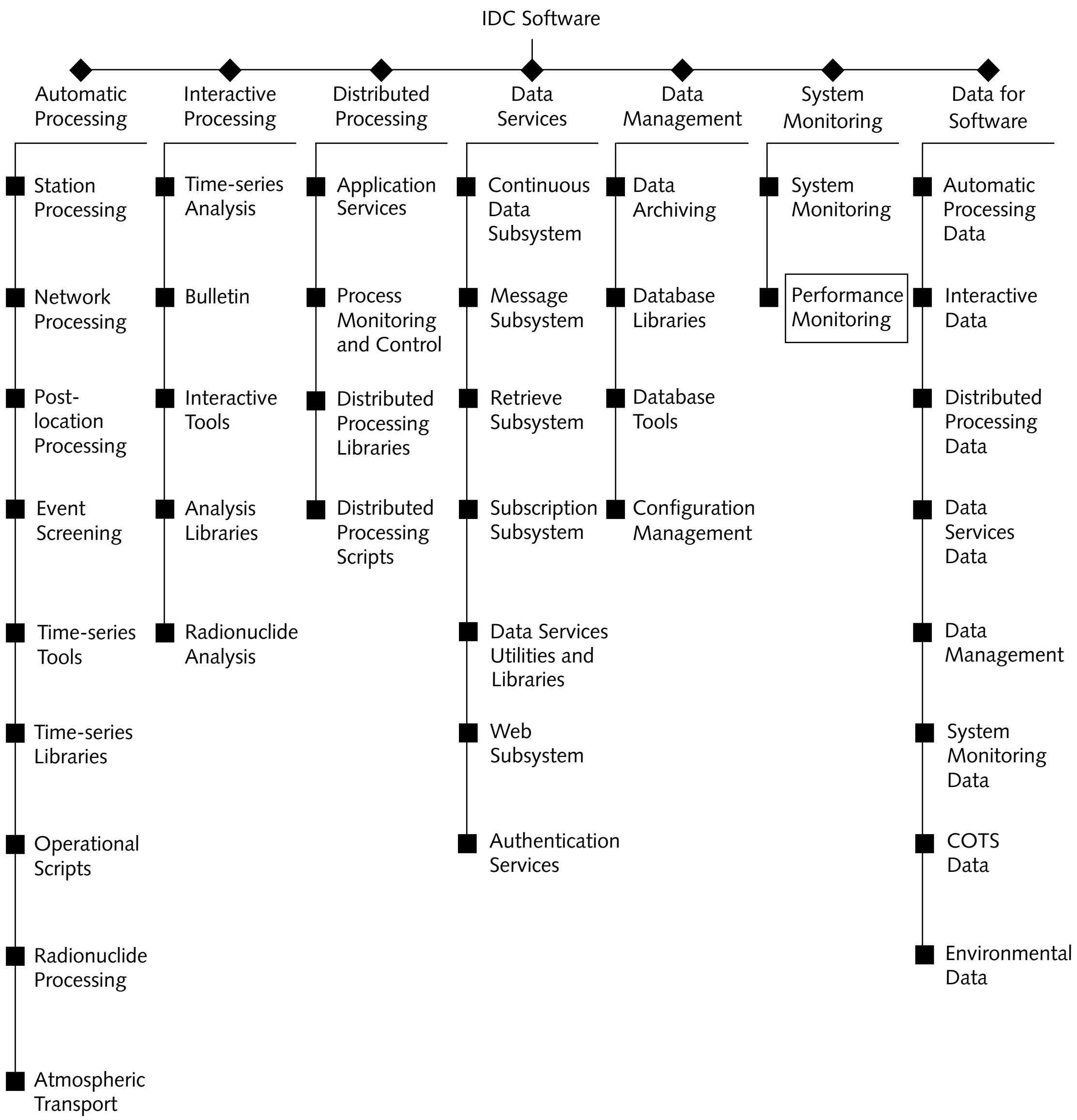


FIGURE 1. IDC SOFTWARE CONFIGURATION HIERARCHY

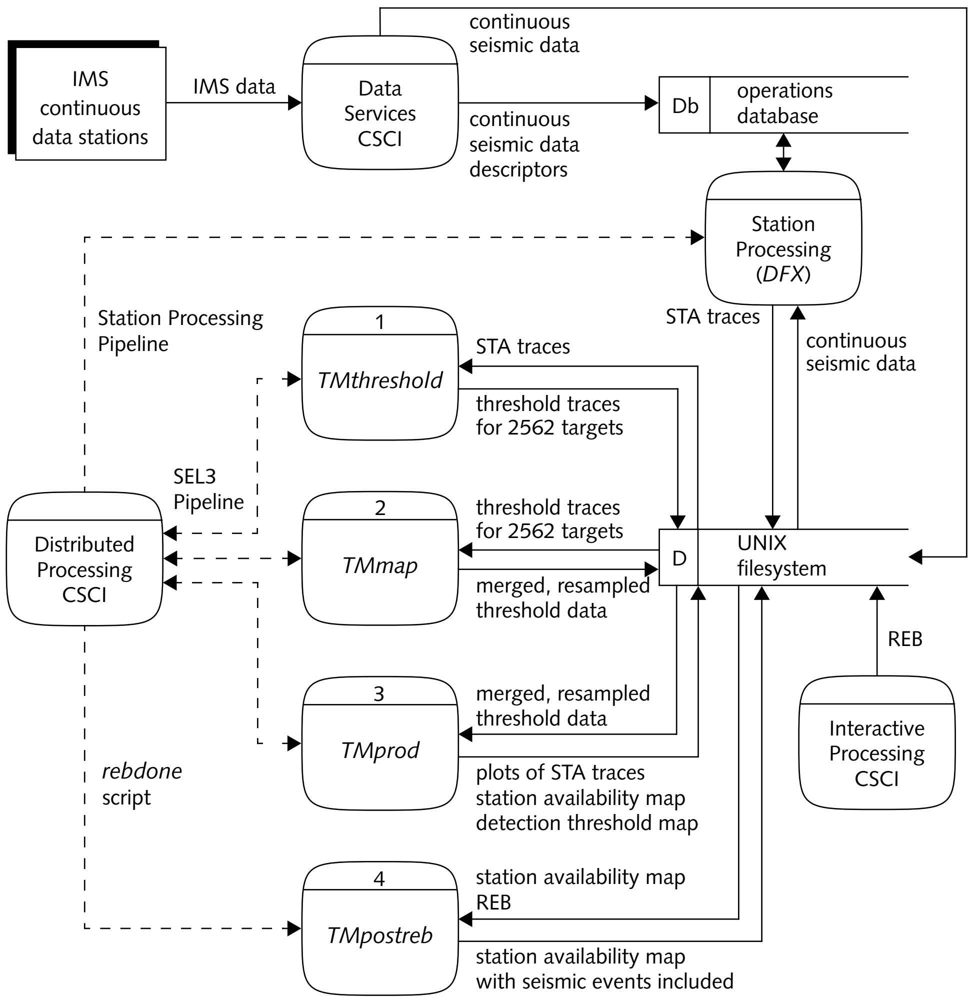


FIGURE 2. TM PROCESSING FLOW

To facilitate map visualization of the results, the *TMmap* program interpolates, reformats, and writes threshold traces for each time sample to a new disk loop (process 2 in Figure 2). *TMmap* is run in the SEL3 pipeline after *TMthreshold* has completed the detection threshold calculations for the given time segment.

Following the STA trace and network detection threshold calculations, three sets of hourly results from TM are calculated by a Bourne shell script called *TMprod* (process 3 in Figure 2), which is run in the SEL3 pipeline.

When the Reviewed Event Bulletin (REB) is complete for a given day, the Bourne shell script *TMpostreb* recreates data availability maps to include information on interfering events (process 4 in Figure 2). This script is invoked in the *rebdone* script.

STATUS OF DEVELOPMENT

TM was developed by NORSAR, a Norwegian research foundation having a long history of contributions within test-ban monitoring. NORSAR has been designated as the Norwegian National Data Center (NDC) for Comprehensive Nuclear-Test-Ban Treaty (CTBT) verification. Version 3.4 of TM includes a few changes to *CreateTMSession*, *AddTMStation*, *DeleteTMStation*, and *LoopCopy*. One print statement was changed in *TMmap*, and the *ctms* library was updated. Code for formerly "external" libraries is included. The static data directory in version 3.4 contains new station processing recipes for ARCES, NVAR, and PDYAR. Future versions may be released with new station processing recipes.

FUNCTIONALITY

The function of TM is to monitor compliance with the CTBT, in particular by placing confidence in the performance of the primary seismic network of the International Monitoring System (IMS), but also by giving a warning when monitoring capability is lowered, for example, by station outages, communication problems, data processing problems, or extremely high seismic activity.

The four main processing programs in TM are *TMthreshold*, *TMmap*, *TMprod*, and *TMpostreb*. Several other programs are called by the main programs. Functional descriptions of the four main programs and the programs that they call follow:

TMthreshold reads STA disk loops, estimates thresholds at 2,562 target points, and writes to the TM disk loops

TMmap	reads TM disk loops for each target, interpolates/resamples the data and writes the data (in Regional Data Format [RDF]) to the `grid.rdf` file (calls *makecdf.2562*)
makecdf.2562	makes a `netCDF` file from the 2,562 target points (called by *TMmap*; calls the Generic Mapping Tools [GMT] script *surface*)
TMprod	analyzes the output from the TM subsystem and plots hourly statistics (calls *tmstatus*, *plotuptime*, and *detplot*)
tmstatus	creates the TM status plot (called by *TMprod*; calls *tm_stast*)
tm_stast	analyzes and plots STA traces (called by *tmstatus*)
plotuptime	plots a world map with all of the primary seismic stations in the IDC network, color-coded to show the percentage of time they were functioning (called by *TMprod* and *loopuptime*)
detplot	creates detection capability plots (called by *TMprod*; calls *rdf2cdf*)
rdf2cdf	reads data from the `grid.rdf` disk loop and writes a `netCDF` file for use by GMT in creating a map (called by *detplot*)
TMpostreb	adds event information to the maps produced by *plotuptime* (calls *TMbulletin* and *replotuptime*)
TMbulletin	retrieves entries from the REB (called by *TMpostreb*)
replotuptime	checks the last processed day in the Late Event Bulletin database account ($LEBDB) (called by *TMpostreb*; calls *loopuptime*)
loopuptime	executes *plotuptime* repeatedly within the given time interval (calls *plotuptime*; called by *replotuptime*)

Figure 3 shows the TM processing programs and their relationships.

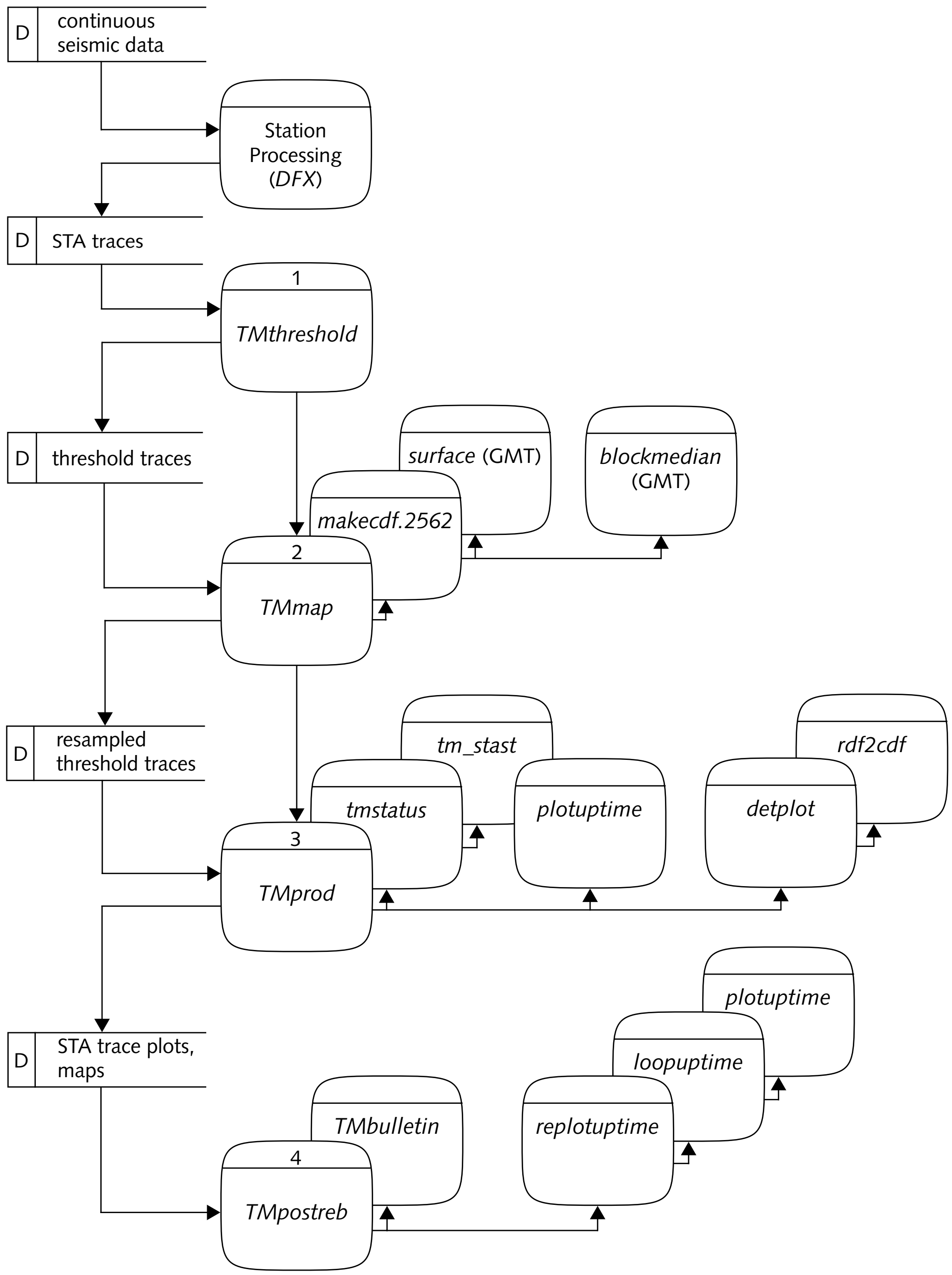


FIGURE 3. TM PROCESSING HIERARCHY

TMthreshold calculates network detection thresholds for each target point. Extensive information on how to combine the continuous STA data from each of the primary seismic stations is required to make this calculation. Recipe files containing the information are created during the setup of the TM processing environment using *CreateTMSession* (see "CreateTMSession Output" on page 106).

TMmap uses the GMT functions *blockmedian* and *surface* [Wes95] to interpolate and resample the detection thresholds. The results are stored in a disk loop where the interpolated detection thresholds for each time sample (at 10-second intervals) are stored sequentially.

Before calculating the hourly statistics, *TMprod* checks the end-time of the processing segment for the completion of a full hour. If a full hour is not completed, *TMprod* does not process.

TMprod calls three other Bourne shell scripts in sequence:

1. The *tmstatus* script reads and analyzes the STA data for each of the stations. It creates an hourly plot of the type shown in Figure 5 on page 12. This plot provides an overview of the background noise levels and the observed signals at each of the primary stations. In addition, *tmstatus* calculates the statistics on available and processed data from the STA disk loops, where NULL values indicate data gaps and processing gaps. A file with the statistics on station availability is made available to the next program in the processing sequence, *plotuptime*.
2. The *plotuptime* script creates results of the type shown in Figure 4 on page 11. These results provide information on the data availability and interfering events given in the REB for the particular 1-hour interval. When *plotuptime* is called in the SEL3 pipeline, the REB is usually not complete. Therefore, initially, event information is not on these plots.
3. The *detplot* script analyzes the interpolated detection thresholds from *TMmap* to provide global information on the average and poorest detection capabilities for the analyzed hour, as shown in Figure 6 on page 13.

The results generated in *TMprod* are stored as PostScript files, and the information related to each plot (time interval, filename, directory name, plot type) is stored in the **fileproduct** database table for subsequent retrieval.

When the REB is complete for a given day, the Bourne shell script *TMpostreb* recreates results of the type shown in Figure 4 on page 11 to include information on interfering events. *TMpostreb* calls several Bourne shell scripts. If the REB is complete for a new day, *plotuptime* is executed for each one-hour interval during the given day. *TMbulletin* reads the REB entries, and *replotuptime* checks the latest time for the bulletin. If a day has been completed, *loopuptime* re-executes *plotuptime* for each hour of that day.

Features and Capabilities

TM generates three sets of results on an hourly basis. These quantify the network detection capability of the primary seismic network and provide information on factors causing a possible degradation of the detection capability. Figures 4, 5, and 6 show results from TM, which describe the network detection capability for a one-hour interval.

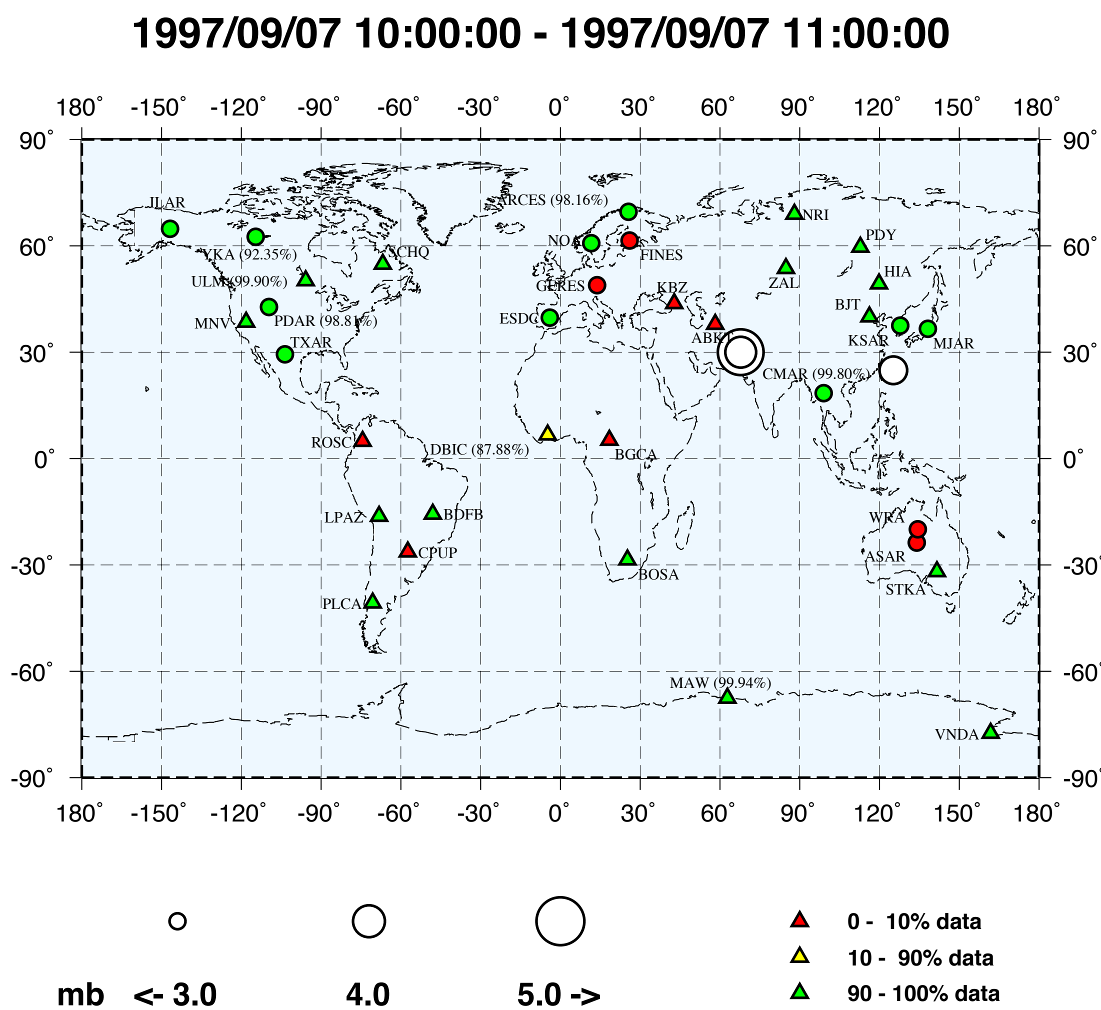
The first set of results (Figure 4) provides information on data availability and interfering events for the one-hour interval. The colors of the station symbols provide information on the availability of data for the interval. Arrays are represented as circles, and three-component (3-C) stations are represented as triangles. Red symbols indicate that data were successfully recorded and processed less than 10 percent of the total time interval; yellow symbols indicate a success rate between 10 percent and 90 percent; and green symbols indicate that for more than 90 percent of the time data were successfully recorded and processed.

For the interval reported, the Australian arrays (ASAR and WRA), the GERES and FINES arrays in Europe, BGCA, and partly DBIC in Africa were all down.

When the REB is complete, the locations of events originating within or close to the actual time interval are plotted. Event information is given below the map.

The second set of results provides an overview of the background noise levels and the observed signals at each of the primary stations during the data interval (Figure 5 on page 12). The traces shown are continuous log(A/T) equivalents derived from the STA traces. Signals from a m_b 4.9 event in Pakistan are seen at most stations in the primary network. The STA traces are calculated from filtered beams for the arrays and from filtered vertical-component channels for the 3-C stations. For each station, the cutoffs of the filter bands used for teleseismic monitoring are given below the station labels along with the average values of the log(A/T) equivalents (which are an overall measure of the background noise level) and the percentages of successfully recorded and processed data. Intervals with gaps in data processing are shown in red above the time axis.

The third type of result provides information on network detection capability (Figure 6 on page 13). The detection capability from TM approximates the 90 percent probability of detecting P waves by at least three stations. The upper map of Figure 6 shows the average network detection capability for the hour. Variation of the average detection capability from hour to hour is primarily caused by differences in long-duration station or processing outages, by changing background noise levels at the different stations, or by long-duration signals from large seismic events. The low detection capability around Australia and Africa is caused by the outages of stations ASAR, WRA, BGCA, and partly DBIC. The lower map of Figure 6 shows the poorest detection capability for the same hour. The differences between this map and the average capability map are primarily caused by signals from seismic events, short outages, and data errors such as unmasked spikes or electronic noise. The m_b 4.9 event in Pakistan temporarily lowers the detection capability all over the world, particularly in the vicinity of the actual event location.



IDC REB

Orid	Time	Lat	Lon	Depth	mb	ndef
1127093	10:15:24	29.97	67.78	19.6	4.88	60
1127104	10:41:00	29.98	67.78	0.0	3.93	14
1127772	11:17:02	24.90	125.13	39.8	3.76	20

FIGURE 4. TM DISPLAY OF DATA AVAILABILITY AND INTERFERING EVENTS

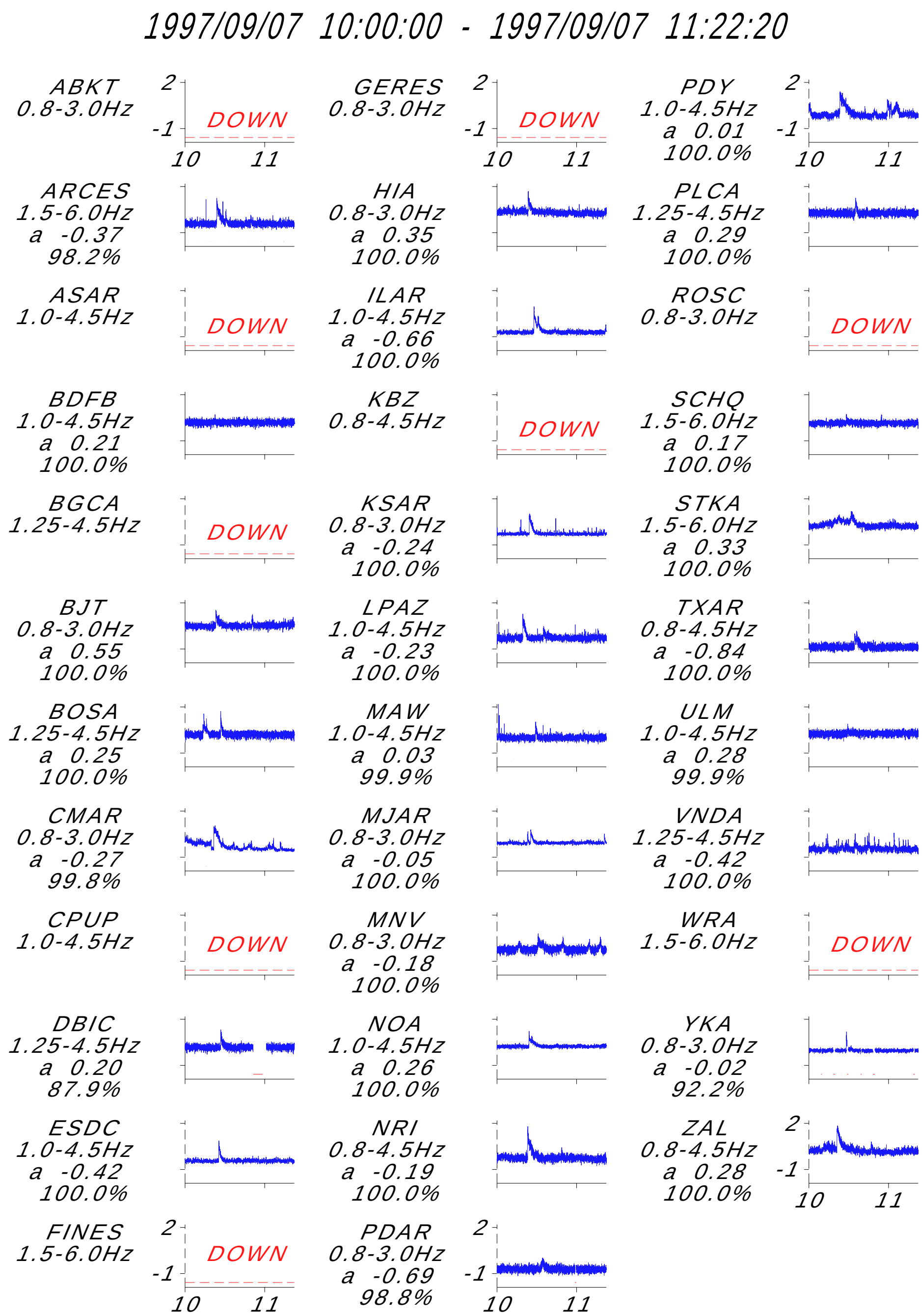


FIGURE 5. TM DISPLAY OF BACKGROUND NOISE LEVEL AND SIGNALS

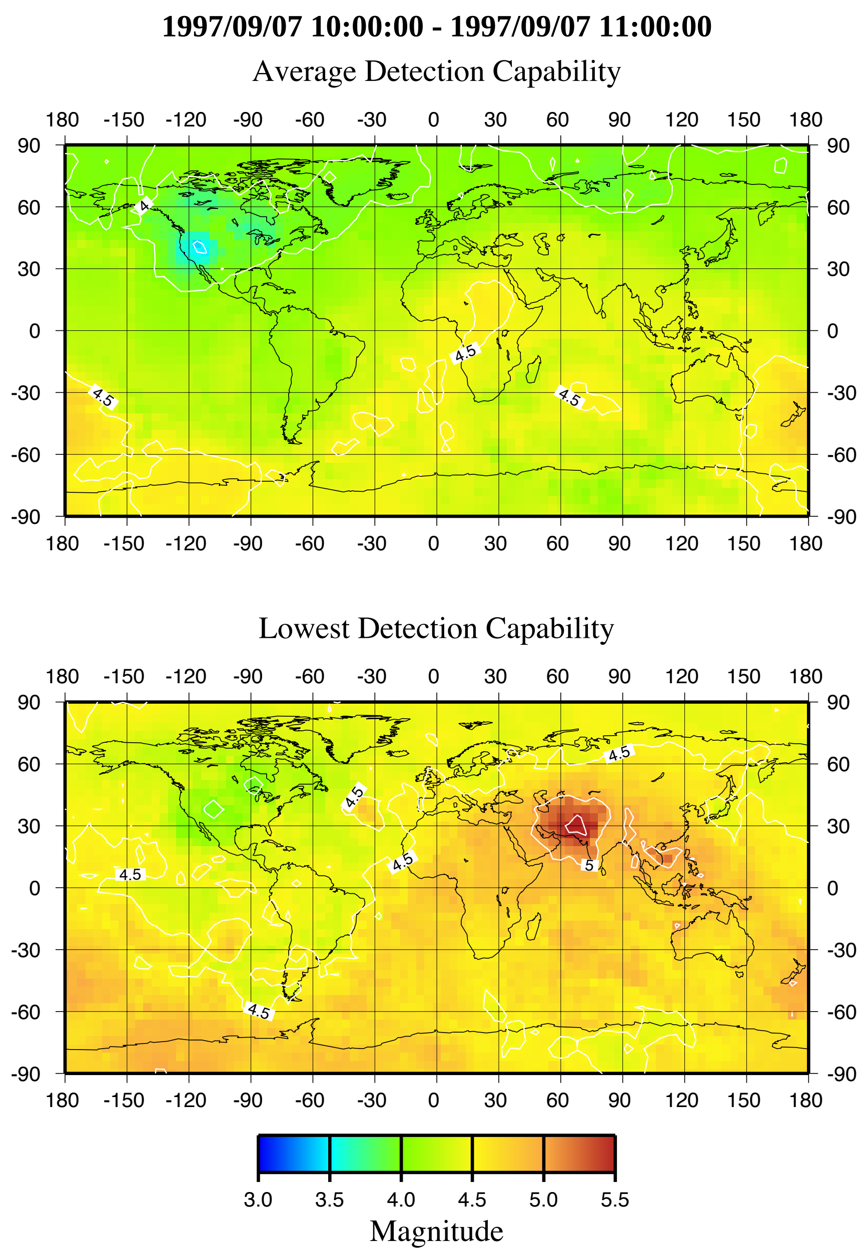


FIGURE 6. TM DISPLAY OF AVERAGE AND POOREST NETWORK DETECTION CAPABILITY

Both types of maps shown in Figure 6 provide important information on the capability of the primary seismic network to detect events in different parts of the world. The information provided in Figure 4 on page 11 and Figure 5 on page 12 help to explain the variations in the global event detection capability.

Performance Characteristics

TM performance is dependent on the number of stations that are included in the calculations and the number of beams that are calculated for each station. The larger the number of stations/beams, the more processing is required to accomplish the task. TM must write to a locally mounted disk due to the amount of input/output (I/O) performed; otherwise, the performance of TM is significantly retarded.

In practice, TM has been reliable 99.99 percent of the time.

Related Tools

TM includes five utilities: *CreateTMSession*, *AddTMStation*, *DeleteTMStation*, *LoopCopy*, and *CopyPSFile*. The first three utilities are used to set up and alter a TM session; the last two copy data. TM also includes scripts to compile the libraries and programs. These utility programs, the programs that they call, and the compilation scripts are described as follows:

CreateTMSession	creates the directory structure and initializes files for a new TM session (calls *tm_beambasic* and *tm_globrec*)
AddTMStation	adds new stations to an existing TM session (calls *tm_beambasic* and *tm_globrec*)
DeleteTMStation	removes stations from an existing TM session (calls *tm_globrec*)
tm_beambasic	generates TM beam recipes for each station (called by *CreateTMSession* and *AddTMStation*)

tm_globrec	generates TM beam recipes for each target (called by *CreateTMSession*, *AddTMStation*, and *DeleteTMStation*)
LoopCopy	extracts data from disk loops
CopyPSFile	copies its argument to `${PSFILE}`
makelibs	compiles TM software libraries
makemods	compiles and links TM software

INVENTORY

The files listed in this section include all of the software and parameter files delivered with TM. Some parameter files (DACS and *DFX* software) are not part of the TM delivery, but must be edited for proper operation of TM. Existing DACS and *DFX* parameter files that must be edited are described in "Installation Procedures" on page 75.

Software

The software structure described in this section reflects the installation at the Prototype International Data Centre (PIDC). The environment variable `$CMS_HOME` is set to `/cmss/rel`.

The directory `$CMS_HOME/src/automatic/src/TM/` contains the TM software. It includes the following subdirectories:

`bin/`	TM executables (compiled from `src/`)
`doc/`	TM manual and related files
`lib/`	TM subroutine libraries (compiled from `libsrc/`)
`libsrc/`	source code (FORTRAN and C) for TM libraries

`scripts/`	shell scripts needed during execution of TM
`src/`	source code for TM programs (FORTRAN and C)

`$CMS_HOME/src/automatic/src/TM/` also contains the scripts *makelibs* and *makemods*, which compile the software.

src and bin

The directory `$CMS_HOME/src/automatic/src/TM/src/` contains a subdirectory for each main program in TM. Each subdirectory contains a `Makefile` in addition to the source and object files. However, the *makemods* script compiles and links all programs. `$CMS_HOME/src/automatic/src/TM/bin/`contains the executable programs. As an example, the executable *TMmap* is found in the `bin/` directory, and the source and object files (`TMmap.c` and `TMmap.o`) are found in the `src/TMmap/` directory.

The `bin/` directory contains the following executable files:

AddTMStation	C program (source file is `addstn.c`)
CreateTMSession	C program
DeleteTMStation	C program (source file is `delstn.c`)
LoopCopy	C program (source file is `loopcopy.c`)
TMmap	FORTRAN subroutines and C main program in `src/`
TMthreshold	FORTRAN subroutines in `libsrc/`, called by the C program in `src/`
rdf2cdf	FORTRAN subroutines in `libsrc/`, called by the C program in `src/`
tm_beambasic	FORTRAN program

tm_globrec	FORTRAN program
tm_stast	FORTRAN program

libsrc and lib

Source and object code for libraries containing C and FORTRAN subroutines for TM are found in the subdirectories of `$CMS_HOME/src/automatic/src/TM/libsrc/`. The subdirectories are named after the libraries they contain (for example, `TM/libsrc/ci/` contains the source and object files for the *ci* library). Each subdirectory also contains a *Makefile*. However, the *makelibs* script described in "Related Tools" on page 14 compiles all libraries.

The compiled libraries in `$CMS_HOME/src/automatic/src/TM/lib/` are named `lib`<*name*>`.a` (for example, `libci.a`). These are linked to the main TM programs by *makemods*.

The libraries are described as follows:

`libci.a`	FORTRAN command interpretation utilities
`libctms.a`	C functions used by *CreateTMSession*, *AddTMStation*, *DeleteTMStation*, *TMmap*, *TMthreshold*, and *LoopCopy* (no scientific algorithms)
`libepo.a`	FORTRAN subroutines for time conversions
`libgmt.a`	GMT software (special version with only the necessary code for TM, adapted to be callable from within a program)
`libgo.a`	FORTRAN utilities related to coordinate conversions and travel times (some files contain multiple functions and subroutines for array manipulation and interpolation, string processing, and I/O)
`libmag.a`	FORTRAN magnitude, azimuth, and travel-time table utilities

`libnetcdf.a`	C utilities for interfacing with FORTRAN, I/O, and so on
`libngbase.a`	utilities (including plotting) from the NORSAR Graphics product line
`libngut.a`	utilities (file handling and type conversions) from the NORSAR Graphics product line
`librdfio.a`	FORTRAN code for creating, reading, and writing the `grid.rdf` disk loop (`rdfio.f` contains the source code for all subroutines and functions)
`libtmcf.a`	FORTRAN code for creating, reading, and writing the disk loops containing the STA and TM data
`libtmdir.a`	FORTRAN TM recipe utilities

scripts

The directory `$CMS_HOME/src/automatic/src/TM/scripts/` contains Bourne shell scripts that are executed in the pipelines as part of TM. The scripts in this directory include *CopyPSFile*, *TMbulletin*, *TMpostreb*, *TMprod*, *detplot*, *loopuptime*, *makecdf.2562*, *plotuptime*, *replotuptime*, and *tmstatus*.

Parameters

The parameter file structure described in this section reflects the installation at the PIDC. Descriptions of all parameters are given in "Installation Procedures" on page 75. The environment variable `$CMS_CONFIG` is set to `/cmss/config`, and `$TM_HOME` is set to `/cmss/config/app_config/monitoring/TM`.

DACS and TM

The DACS parameter files for the TM programs *TMthreshold*, *TMmap*, *TMprod*, and *TMpostreb* are found in the directory `$CMS_CONFIG/app_config/distributed/tuxshell/tm/` and include `tuxshell-TMthresh.par`, `tuxshell-TMmap.par`, `tuxshell-TMprod.par`, and `tuxshell-TMpr.par`, respectively. Parameter descriptions are given in "Tuxedo Files" on page 99.

The `$TM_HOME` directory contains parameter files for the programs *CreateTMSession*, *TMthreshold*, and *TMmap*. These parameter files are `CreateTMSession.par`, `TMthreshold.par`, and `TMmap.par`, respectively. Parameter descriptions are given in "Initiating Operations" on page 103.

DFX

One TM-specific par file is needed by *DFX*: `$CMS_CONFIG/app_config/DFX/tm/tm.par`. Other *DFX* par files must be configured for TM processing as described in "DFX" on page 79.

Static Data

To create and run a processing session, a set of permanent data files must be available. These files reside in a directory structure <*staticdir*> (normally `$TM_HOME/data/`) as shown in Figure 7. Descriptions of the files and their contents are given in "Static Data" on page 87.

```
<staticdir>=$TM_HOME/data/─┬─ Targets/ ── targets.2562
                           │
                           ├─ Beamset/ ── <stn>/ ──────┬─ beambasic
                           │                           └─ missteer.<dhig>-<dlow>
                           │
                           ├─ libdata/ ─┬─ TM.site
                           │            │
                           │            ├─ magtabl/ ──── tab.PTM
                           │            │
                           │            └─ trtabl/ ───── tab.PTM
                           │
                           └─ par/ ─────┬─ detplot/ ───┬─ data.cpt.2
                                        │              ├─ data.cpt.3
                                        │              └─ .gmtdefaults
                                        │
                                        ├─ tmstatus/ ──┬─ ACRES.traces
                                        │              ├─ ASAR.traces
                                        │              ├─ CMAR.traces
                                        │              ├─ ESDC.traces
                                        │              ├─ ILAR.traces
                                        │              ├─ KSAR.traces
                                        │              ├─ MJAR.traces
                                        │              ├─ NOA.traces
                                        │              ├─ NORES.traces
                                        │              ├─ PDAR.traces
                                        │              ├─ TXAR.traces
                                        │              ├─ TXAR.traces
                                        │              ├─ WRA.traces
                                        │              ├─ YKA.traces
                                        │              └─ tmstatus.dat
                                        │
                                        └─ uptime/ ────┬─ cur_pri.data
                                                       ├─ offset.data
                                                       └─ .gmtdefaults
```

FIGURE 7. STATIC DATA DIRECTORY

Fonts and Color Tables

The directory `$TM_HOME/aux/` contains the fonts and color tables used by *tm_stast*. The subdirectory `$TM_HOME/aux/fonts/` contains one file: `nogra.fnt`. The following files are found in `$TM_HOME/aux/colors/` (all fonts and color tables are part of the NOGRA [NORSAR Graphics] product line):

```
Basic.Ngcol               Palette_Basic.Ngcol
Blue_Scale.Ngcol          Red_Scale.Ngcol
Blue_White_Red.Ngcol      Red_White_Blue.Ngcol
Cold_Warm.Ngcol           Rose_Scale.Ngcol
```

```
Cyan_Scale.Ngcol          Violet_Scale.Ngcol
Gold_Scale.Ngcol          Warm_Cold.Ngcol
Green_Scale.Ngcol         Yellow_Scale.Ngcol
Palette.Ngcol
```

ENVIRONMENT AND STATES OF OPERATION

Software Environment

TM uses the *DFX* software to calculate continuous STAs using data from the primary seismic stations of the IMS. The GMT software [Wes95] is used to produce maps for the products produced by TM. Control of TM is maintained through the DACS [IDC6.5.2Rev0.1].

A disk with at least 2.5 GB of free space is needed for TM processing, although 4 GB is recommended to accommodate future expansion of the primary seismic network or any increase in the length of the files written. Due to the amount of I/O performed, TM must write to a locally mounted disk; otherwise, the performance of TM is significantly retarded.

One year's worth of output files from TM requires about 4.7 GB. Sufficient disk space should also be available for long-term storage of these files.

Normal Operational State

Normally, TM is invoked by the station processing pipeline, the SEL3 pipeline, and the *rebdone* script. Pipelines are controlled by the DACS, and the *rebdone* script is initiated manually when the bulletin review has been completed. A data flow diagram of TM is shown in Figure 8. All of the inputs and outputs for the main TM programs are shown in the figure.

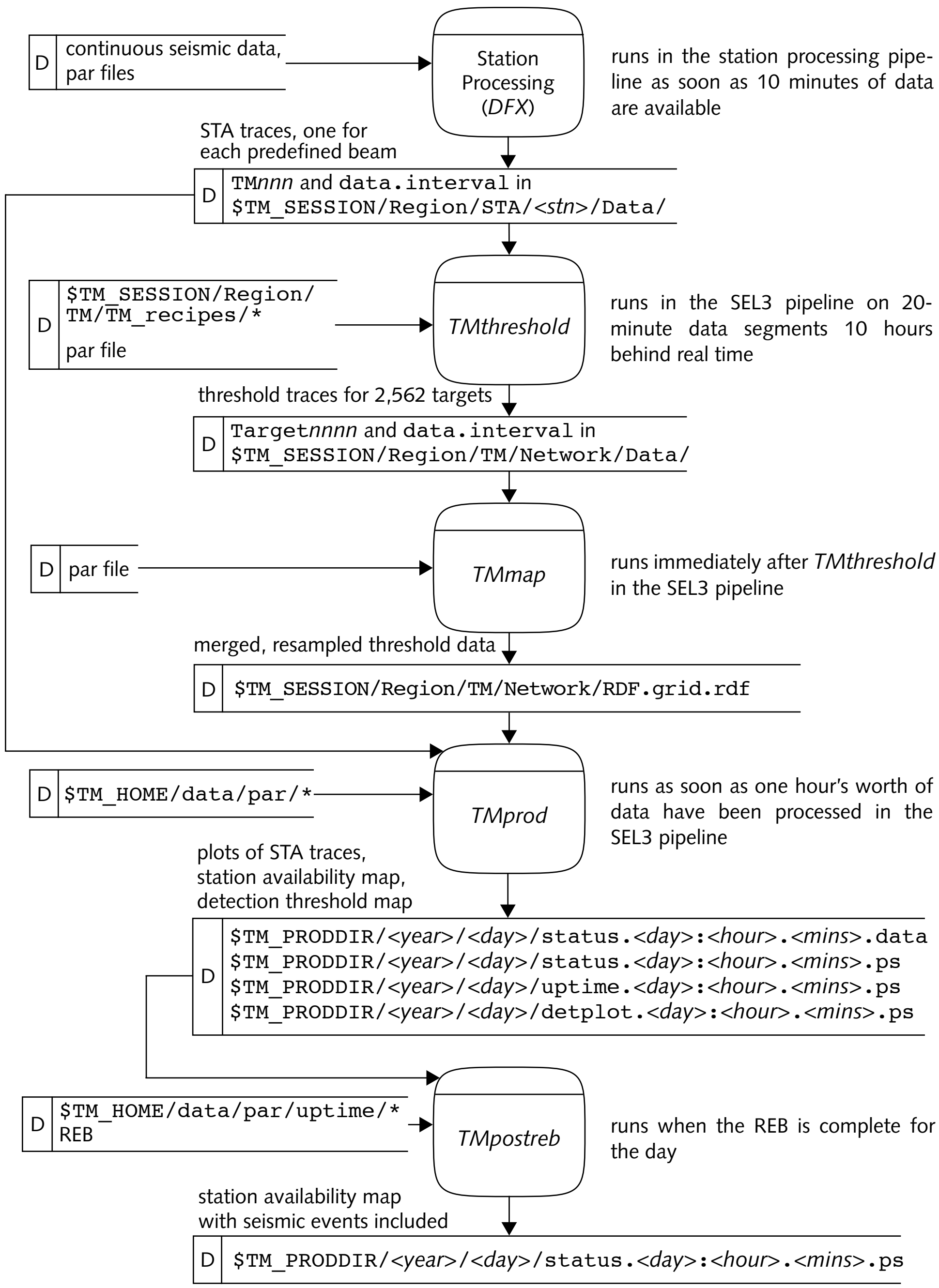


FIGURE 8. TM DATA FLOW

Contingencies/Alternate States of Operation

TM may be executed separately from the pipeline operations for testing (perhaps with a different station list) or tuning. Several of the programs are designed specifically for setting up a processing session (for example, *CreateTMSession*) or altering the parameters of the existing session (*AddTMStation*, *DeleteTMStation*).

When testing or tuning, you may want to use a static data set or smaller versions of the station disk loops (covering a more restricted time span). *LoopCopy* is a utility provided to copy disk loops.

Chapter 2: Operational Procedures

This chapter provides instructions for using the software and includes the following topics:

- Software Startup
- Software Shutdown
- Basic Procedures
- Advanced Procedures
- Maintenance

Chapter 2: Operational Procedures

SOFTWARE STARTUP

Most of the TM programs are controlled by the DACS as part of the station processing and SEL3 pipelines (see [IDC6.2.1]). *TMpostreb* is controlled by the *rebdone* script. However, you can start all programs of the TM subsystem manually.

To run TM manually (the processing environment must be established using *CreateTMSession*):

1. Log onto the machine on which *DFX* and the TM programs are run:

   `% rlogin` <*machine*>

2. Run *DFX* using the following command line:

   `% DFX par=dfx.par`

3. Run *TMthreshold* using a command line in the following format:

   `% TMthreshold par=`<*parfile*> `t1=`<*time1*> `t2=`<*time2*> `[verbose]`

   Specify the times of interest, <*time1*> and <*time2*>, either in epoch time or in the format *YYYY-DDD:HH.MM.SS.sss*.

   Create recipe files for *TMthreshold* during setup of the TM processing environment (see "Creating TM Sessions" on page 29). Input parameters are either read from the command line or from a parameter file (see "TMthreshold" on page 85 for a description of the parameter file).

   The following example command line runs *TMthreshold* for a 20-minute time interval:

   ```
   % TMthreshold par=TMthreshold.par t1=1998-040:02.40\
   t2=1998-040:03.00 verbose
   ```

4. Run *TMmap* using a command line in the following format:

   `% TMmap par=`*<parfile>* `t1=`*<time1>* `t2=`*<time2>* `[verbose]`

   You can specify the times of interest, *<time1>* and *<time2>*, either in epoch time or in the format *YYYY-DDD:HH.MM.SS.sss*.

   Input parameters for *TMmap* are either read from the command line or from a parameter file (see "TMmap" on page 86 for a description of the *TMmap* parameters).

   The following example command line runs *TMmap* for a 20-minute time interval:

   ```
   % TMmap par=TMmap.par t1=1998-040:02.40\
   t2=1998-040:03.00 verbose
   ```

5. Run *TMprod* using a command line in the following format:

   `% TMprod` *<Session> <productdir> <t1> <t2> <database>*

   TMprod, a Bourne shell script, does not use the *getpar* command line parser, so the arguments to *TMprod* are regular arguments to a shell script and cannot contain a par file. *<Session>* is the fully specified path for the TM processing environment (`$TM_SESSION`). *<productdir>* is the fully specified path for files containing hourly results (`$TM_PRODDIR`). *<t1>* is the processing segment start-time in epoch format. *<t2>* is the processing segment end-time in epoch format. *<database>* is the database account containing the **fileproducts** table (should be `none` if the database is not used).

   Establish appropriate environment variables ($TM_SESSION and $TM_PRODDIR) to substitute for *<Session>* and *<productdir>* (see "Environment Variables" on page 77).

   The following example command line runs *TMprod*:

   ```
   % TMprod $TM_SESSION $TM_PRODDIR 886992000 886993200\
   none
   ```

   The following example command line has a database account containing the **fileproducts** table:

   ```
   % TMprod $TM_SESSION $TM_PRODDIR 886992000 886993200\
   $IDCXDB
   ```

6. Run *TMpostreb* using a command line in the following format:

   `% TMpostreb` <*Session*> <*productdir*> <*rebdb*> <*database*>

   TMpostreb, a Bourne shell script, does not use the *getpar* command line parser, so the arguments to *TMpostreb* are regular arguments to a shell script and cannot contain a par file. <*Session*> is the fully specified path for the TM processing environment (`$TM_SESSION`). <*productdir*> is the fully specified path for files containing hourly results (`$TM_PRODDIR`). <*rebdb*> is the name of the database account containing REB entries. <*database*> is the name of the database account containing the **fileproducts** table with the time for the latest REB entry.

   Establish appropriate environment variables (for example, $TM_SESSION, $TM_PRODDIR, $REBDB, and $IDCXDB) to substitute for all of these parameters (see "Environment Variables" on page 77). For example, the following *TMpostreb* command uses environment variables as parameters:

   ```
   % TMpostreb $TM_SESSION $TM_PRODDIR $REBDB $IDCXDB
   ```

SOFTWARE SHUTDOWN

Under normal operations, TM is invoked as part of the operations pipelines. Instructions for shutting down pipeline operations are described in [IDC6.2.1].

BASIC PROCEDURES

TM has no operator control interface. Basic procedures consist of starting up and shutting down the station processing and SEL3 pipelines through the DACS.

ADVANCED PROCEDURES

This section provides detailed instructions for using the software's advanced features.

Creating TM Sessions

The directory structure and most of the files required to run TM are created by *CreateTMSession*. *DFX* must have write privileges in <*Session*> for TM to work.

1. Log on using an account that is accessible by *DFX*, or start a shell with an appropriate group identification (ID) (for example, with the UNIX command `newgrp`).
2. Run *CreateTMSession* with a command line in the following format:

   `% CreateTMSession par=<parfile> [verbose[=n]]`

   The <*parfile*> entries are described in Table 20 on page 104. *CreateTMSession* has optional "verbose" modes that cause informative messages to be printed out as the program runs. For example, the `verbose` option displays the name of each file and directory created (except for the disk loops). A few informative messages are also printed from some of the FORTRAN programs and subroutines that are called by *CreateTMSession*. `verbose=2` prints more information, including the name of each disk loop file created (of which there may be thousands). Additional information from the FORTRAN programs (primarily useful for debugging) is printed at higher verbosity levels.

   Any of the parameters in the parameter file may appear on the command line. Exactly the same format is used. The search order is from left to right on the command line, with precedence given to parameters found later if there are multiple specifications. Parameters in parameter files may be overridden by placing them after the `par=<parfile>` specification.

Adding Stations to Existing Session

The procedure for adding a new station to pipeline TM processing follows. *AddTMStation* only alters the contents of <*Session*>. Other files must be edited manually.

The procedure for including a new station in TM assumes that the following processing parameters are available (see "Tuning Processing Parameters" on page 36):

- prefilter cutoffs for the different distance ranges and the corresponding average differences between log(A/T) and log((π/2)·STA·calib) (see "Set Prefilter Parameters" on page 39)
- information on signal loss as a function of mis-steering of array beams (see "Estimate Signal Loss and Mis-steering" on page 43) and the corresponding beam deployment (see "Establish Beam Set" on page 46)

To include a new station that has no history of event recordings, adopt the processing parameters of a station assumed to have similar noise and signal characteristics.

To add a new station to an existing session:

1. Stop all of the TM processes (*TMthreshold*, *TMmap*, *TMprod*, *TMbulletin*, and *replotuptime*). If *DFX* is already running for one of the stations to be included, stop *DFX* processing for this station.
2. Use *DeleteTMStation* to delete stations from <*Session*> that should be overwritten.
3. Edit the *DFX* parameter file `$OPSDIR/static/DFX/beam/detection/`<*stn*>`-beam.par` to include the definition of the beams used for STA calculations (see "DFX" on page 79).
4. Enter the coordinates for the new station in the file `$TM_HOME/data/libdata/TM.site`. The format of `TM.site` can be created by the program *DumpTable*, which creates an American Standard Code for Information Interchange (ASCII) file in a format corresponding to the **site** table of the ORACLE database.
5. Create the directory `$TM_HOME/data/Beamset/`<*stn*>`/` and within it the files `beambasic` and `missteer.`<*Dlow*>-<*Dhig*> (see "Beamset/" on page 88). These files contain information on the beam set and the mis-steering information for TM. The definitions of the beams must correspond with the definitions in `$OPSDIR/static/DFX/beam/`

`detection/`*<stn>*`-beam.par`, which are given to *DFX*. For 3-C stations set the signal loss to 0.0; lines with signal loss as a function of mis-steering are not necessary (see, for example, `$TM_HOME/data/Beamset/BOSA/missteer.0-180`).

6. Add a new entry for the station (in alphabetical order) to the file `$TM_HOME/data/par/tmstatus/tmstatus.par`. For arrays, the file *<stn>*`.traces` (containing the list of STA beams used for teleseismic monitoring) should accompany `tmstatus.par`. A more detailed description of these parameter files is given in "par/" on page 93.

7. Add corresponding new entries for the station (in alphabetical order) to the files `cur_pri.data` and `offset.data` located in the directory `$TM_HOME/data/par/uptime/`. As explained in "par/" on page 93, these files contain information on the positioning of the station symbols and text strings as shown in Figure 4 on page 11. Some testing of *TMprod* is probably necessary to produce good parameter values for `offset.data`.

8. Run *AddTMStation* to include the new station in the TM processing environment as follows:

   `% AddTMStation par=`*<parfile>* `stlist=`*<stn1>*`,`*<stn2>*`,…,`*<stnn>*`\`
   `[verbose[=`*n*`]]`

   You can use the parameter file with *CreateTMSession* for *AddTMStation*. *AddTMStation* requires the *session_directory*, *stlist*, and *staloop* parameters described in Table 20 on page 104. For *AddTMStation*, the *stlist* parameter gives the list of stations to be added to the processing session. If you enter the *stlist* parameter on the command line after `par=`*<parfile>* you do not have to change it in the parameter file. *AddTMStation* also uses *staticdir*, *stasamp*, and *stalen,* but the defaults shown in Table 20 on page 104 are assumed if they are not given. *AddTMStation* has the same (optional) verbosity parameter as *CreateTMSession.*

   AddTMStation compares the list of new stations to the list of existing stations found in `selected.stations` and only adds those that are not duplicates (stations being upgraded were deleted in step 2).

9. Start the *DFX* processing of the new station, and restart the TM processes (*TMthreshold*, *TMmap*, *TMprod*, and *TMpostreb*).

Deleting Stations from Existing Session

You can delete a station from pipeline TM processing. Files outside <*Session*> must be edited manually.

To delete a station from an existing session:

1. Stop all of the TM processes (*TMthreshold*, *TMmap*, *TMprod*, *TMbulletin*, and *replotuptime*). If *DFX* is running for one of the stations to be deleted, stop *DFX* processing for this station.
2. Edit the *DFX* parameter file `$OPSDIR/static/DFX/beam/detection/`<*stn*>`-beam.par` to remove TM beams for each station being removed from processing.
3. Delete the coordinates for the stations being removed from processing in the file `$TM_HOME/data/libdata/TM.site`.
4. Delete the directory `$TM_HOME/data/Beamset/`<*stn*>`/` and within it the files `beambasic` and `missteer.`<*Dlow*>-<*Dhig*>.
5. Delete entries for the stations being removed from the file `$TM_HOME/data/par/tmstatus/tmstatus.par`. For arrays, delete the file <*stn*>`.traces`.
6. Delete entries for the stations being removed from the files `cur_pri.data` and `offset.data` located in the directory `$TM_HOME/data/par/uptime/`.

7. Run *DeleteTMStation* with the *stlist* parameter listing the stations to be removed from processing:

   `DeleteTMStation par=`*<parfile>* `stlist=`*<stn1>*`,`*<stn2>*`,…,`*<stnn>*`\`
   `[verbose[=`*n*`]]`

   You can use the parameter file used with *CreateTMSession* for *DeleteTMStation*. For *DeleteTMStation*, the *stlist* parameter is the list of stations to be deleted from the processing session. *DeleteTMStation* compares the new station list specified by the *stlist* parameter to the old station list in `selected.stations` and only attempts to remove those stations that appear in both lists. The entire *<Session>*`/Region/Tm_types/` directory structure and `selected.stations` are rewritten, and the TM processing recipes in the file *<Session>*`/Region/`*<tmtype>*`/TM_recipes` are recalculated.

8. Restart the TM processes (*TMthreshold*, *TMmap*, *TMprod*, and *TMpostreb*).

Copying Data from Disk Loops

You can copy all or part of the data in any STA, TM, or RDF disk loop to a new file. Using *LoopCopy*, you may enter starting and ending times of interest and the size of the new file in seconds. These three parameters are not required, and the default values are those found in the old file. However, if none of these parameters are used, then the new file is a duplicate of the old file.

Run *LoopCopy* using the command line:

`% LoopCopy par=`*<parfile>* *<mode(s)>* `[update] [verbose[=`*n*`]]`

The *<parfile>* parameters used by *LoopCopy* are described in Table 1.

TABLE 1: LOOPCOPY PARAMETERS

Parameter	Default	Description
session_directory	none (required)	location of the *TM* processing environment
scratchdir	<*Session*>`/scratch`	scratch directory
stlist	all stations	comma-separated list of stations
t1	start-time of data in input file	start-time of the processing segment
t2	end-time of data in input file	end-time of the processing segment
staloop	*t2*–*t1*+1	size of new STA files in seconds
tmloop	*t2*–*t1*+1	size of new RDF or *TM* files in seconds
tmlist	all targets	comma separated list of targets

LoopCopy expects at least one <*mode*> parameter. The <*mode*> parameter should be:

- `STA` when copying from the STA disk loops in <*Session*>`/Region/STA/`<*stn*>`/Data/` for a list of stations
- `TM` when copying from the TM disk loops in <*Session*>`/Region/TM/`<*tmtype*>`/Data/`
- `RDF` for copying from the `grid.rdf` file in <*Session*>`/Region/TM/`<*tmtype*>`/RDF/`

Any of these parameters may be present in the same call to *LoopCopy*. For example:

```
% LoopCopy par=<parfile> STA RDF TM
```

copies the files for all desired stations, all desired TM disk loops, and `grid.rdf`.

Files are read from the appropriate subdirectory in *<Session>* (specified in *<parfile>*), and new files are created in the designated scratch directory. Output filenames have the following formats:

<input_file>`.`*<tmtype>*	(TM mode, for example, `Target_1001.Network`)
<input_file>`.`*<stn>*	(STA mode, for example, `TM001.ARCES`)
`grid.`*<tmtype>*	(RDF mode, for example, `grid.Network`)

You can specify the times *t1* and *t2* in the *<parfile>* either as epoch times or in the format *YYYY-DDD:HH.MM.SS.sss*. Any data in an input file that fall outside the range given by *t1* and *t2* are ignored. Data falling within this range are copied, discarding the oldest data if the new file is too small (due to the *staloop* or *tmloop* given in *<parfile>*).

If verbosity is desired, the same verbose parameter described for the other programs in this chapter is available. For example, to read and print header information about the new files, use the following command:

```
% LoopCopy par=<parfile> <mode(s)> verbose=2
```

Higher verbosity levels may result in lengthy diagnostics from subroutines.

To change the length of the disk loops, you can overwrite the input files using the `update` option. For example:

```
% LoopCopy par=<parfile> RDF update
```

creates a new RDF file in the designated scratch area and then overwrites the original `grid.rdf` file, destroying the original contents. The file `data.interval` is updated. In the case of STA or TM disk loops, `cf.length` is also updated. The file(s) created in the scratch area are removed.

MAINTENANCE

Tuning Processing Parameters

The TM processing parameters for all stations need to be tuned to reliably estimate the detection capability. Tuning events with good signal-to-noise ratios (snrs), preferably occurring at various distances from the station, are needed. If a new station has no history of recording events, adopt the processing parameters of a station assumed to have similar noise and signal characteristics.

To tune TM processing parameters:

1. Collect a set of tuning events (see "Select Tuning Events" on page 37).
2. Search for noise spikes by plotting the average log(STA) versus frequency for all noise samples (see "Determine Signal and Noise Characteristics" on page 37). Avoid frequencies with noise spikes when determining the frequency band with the best snr.
3. Establish distance ranges over which specific frequency bands provide good snr (usually two ranges: a regional distance range and a teleseismic distance range) by plotting snr as a function of frequency and distance.
4. Compare the reference log(A/T) measurements in the 0.8–4.5 Hz filter band with the log((π/2)·STA·*calib*) measurements using several different filter bands (avoiding noise spikes) for the P waves of the reference events in each distance range. Choose the frequency band for which log(A/T) – log((π/2)·STA·*calib*) of the P waves has little scatter and the average noise magnitude (in m_b units) is small.

   Set the following values using the chosen frequency band for each distance range:

   – filter characteristics
     (in `beambasic`, `tmstatus.dat`, and *stn*`-beam.par`)
   – *ATcomp* (the average of log(A/T) – log((π/2)·STA·*calib*))
     (in `missteer.`*Dlow*`-`*Dhig*)

5. Plot signal loss as a function of the mis-steering of the beams for the tuning events (if the station is an array).

   Set the following values in `missteer.`*Dlow*`-`*Dhig* using the chosen frequency band for each distance range:

   – average signal loss for correctly steered beams
     (in `missteer.`*Dlow*`-`*Dhig*)
   – teleseismic signal loss correction in magnitude units (*sloss*/20) plus the magnitude correction (*ATcomp*)
     (in `tmstatus.dat`)
   – sample signal loss values at various mis-steering values
     (in `missteer.`*Dlow*`-`*Dhig*)

6. Establish a beam pattern for each distance range that covers the slowness space of the distance range. Beams must cover the entire area such that the signal loss is nowhere greater than 3 dB.

   Set the beam steering parameters in *stn*`-beam.par`, and `beambasic`.

A tuning example for the Alaskan ILAR array is given in the sections that follow.

Select Tuning Events

The first step in the tuning procedure is to collect a list of suitable events. To do this, search the database for good snr events at various distances and then request the data intervals using *AutoDRM*. To obtain information on the background noise field, request event data segments starting one minute ahead of the P arrival and having a 2-minute duration.

Determine Signal and Noise Characteristics

The TM procedure for estimating the network detection capability resembles the IDC procedure for estimating m_b. At the IDC, a third-order Butterworth filter with a passband between 0.8 and 4.5 Hz is applied to the data prior to estimating signal amplitude and period. The same prefilter is applied prior to generating the STA envelopes.

For some stations the 0.8–4.5 Hz passband may be contaminated by high noise amplitudes. Select a prefilter frequency band as close as possible to the frequency range where the highest snr of the signals is observed. For many stations these frequencies are significantly higher than 0.8 Hz.

To see if there are specific noise peaks that should be avoided, calculate STAs in narrow filter bands for the noise preceding the first P-arrivals. Steer the beams to the azimuths and slownesses of the P-phases. Figure 9 shows the average ILAR log(STA) for all noise segments versus frequency; no noise peaks are observed. For other stations this type of plot reveals clear noise peaks. For example, at CMAR (Thailand) a clear noise peak near 3 Hz is found. An alternative to this type of plot is to make plots of average regular noise spectra.

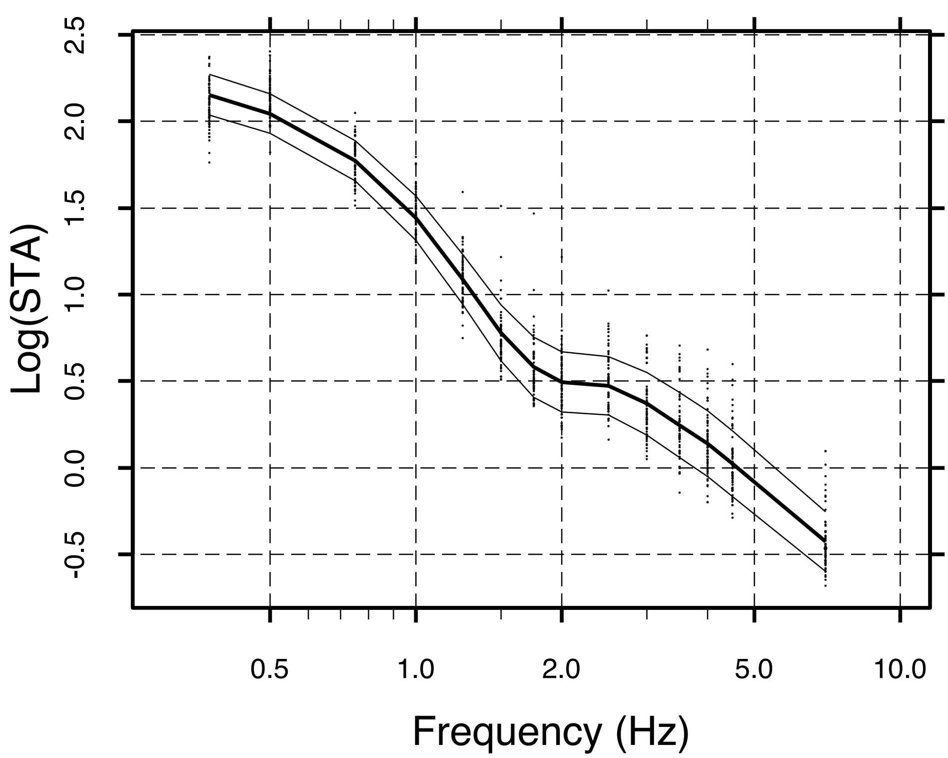


FIGURE 9. AVERAGE NARROWBAND LOG(STA) OF ILAR NOISE SEGMENTS

To find the frequency range in which the highest snr is expected, plot the snr (STA/LTA) measured in narrow frequency bands versus the distance to the events as in Figure 10 for ILAR. In this plot the maximum snr is normalized to 50 dB for each event, and some averaging is done to group the events into distance bins. For events at distances above 15 degrees from ILAR, the highest snr is usually found between 1 and 3 Hz. For events at closer distances, the best snr is moved to somewhat higher frequencies. For the events in the PKP range, the highest snr is between 1 and 2 Hz.

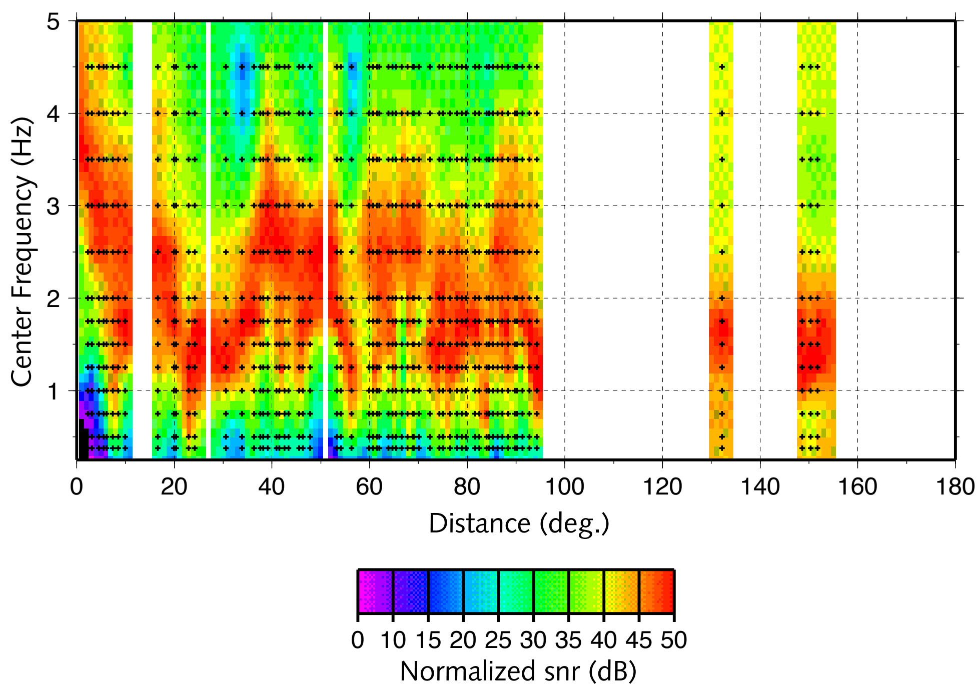


FIGURE 10. SNR (STA/LTA) OF ILAR EVENTS VERSUS EVENT DISTANCE

Set Prefilter Parameters

When choosing the prefilter cutoffs from the tuning events, try to balance low amplitudes during noise conditions with a good recovery of the signal amplitudes. The frequencies may be higher than those used for routine magnitude estimation.

To relate the log(STA) estimates used by TM to the log(A/T) estimates used for magnitude estimation at the IDC, manually measure log(A/T) of the tuning events. Measure A/T for the maximum amplitude occurring within 8 seconds of the first arrival on beams steered with the azimuths and slownesses of the P phases and filtered between 0.8 and 4.5 Hz.

Based on the average noise characteristics (see Figure 9 on page 38) and the frequency range with the highest snr (see Figure 10 on page 39), test a series of filters for subsequent use in TM. The distance varying snr behavior shown in Figure 10 for ILAR indicates that it is reasonable to use different filters for events above and below 15 degrees distance from ILAR. Therefore, the tuning events are split into two groups at this event-station distance.

For events above 15 degrees, Figure 11 shows results of comparing the reference log(A/T) measurements in the 0.8–4.5 Hz filter band with the log((π/2)·STA·*calib*) measurements in three different filter bands (several other filters have also been tested). The average difference between the two measurements is referred to as *ATcomp*. *Mean* and *Median* in the legend of Figure 11 give the mean and median, respectively, of the differences. *St.dev.* gives the standard deviation of the differences, and *Noise* gives the average STA level of the preceding noise.

STAs measured in the 0.8–4.5 Hz filter band have, as expected, a very good correspondence with the reference A/T values. The mean difference is only 0.01 m_b units with a standard deviation of 0.05. However, the average noise value (log((π/2)·STA·*calib*)) of the P-beams in this frequency band is –0.84 (m_b units).

The 1.0–4.5 Hz filter band best combines low amplitudes during noise conditions with a good and stable recovery of the signal amplitudes. When processing ILAR data in this frequency band, 0.09 m_b units (*ATcomp*) is added to the log((π/2)·STA·*calib*) estimates to make them compatible with magnitude estimates at the IDC.

The results for events within 15 degrees are shown in Figure 12. No events occur between 10 and 15 degrees; therefore, the plot stops at 10 degrees. Again, results for three filters are shown, and the 1.25–4.5 Hz filter band is preferred due to its low noise level and its small mean difference and standard deviation.

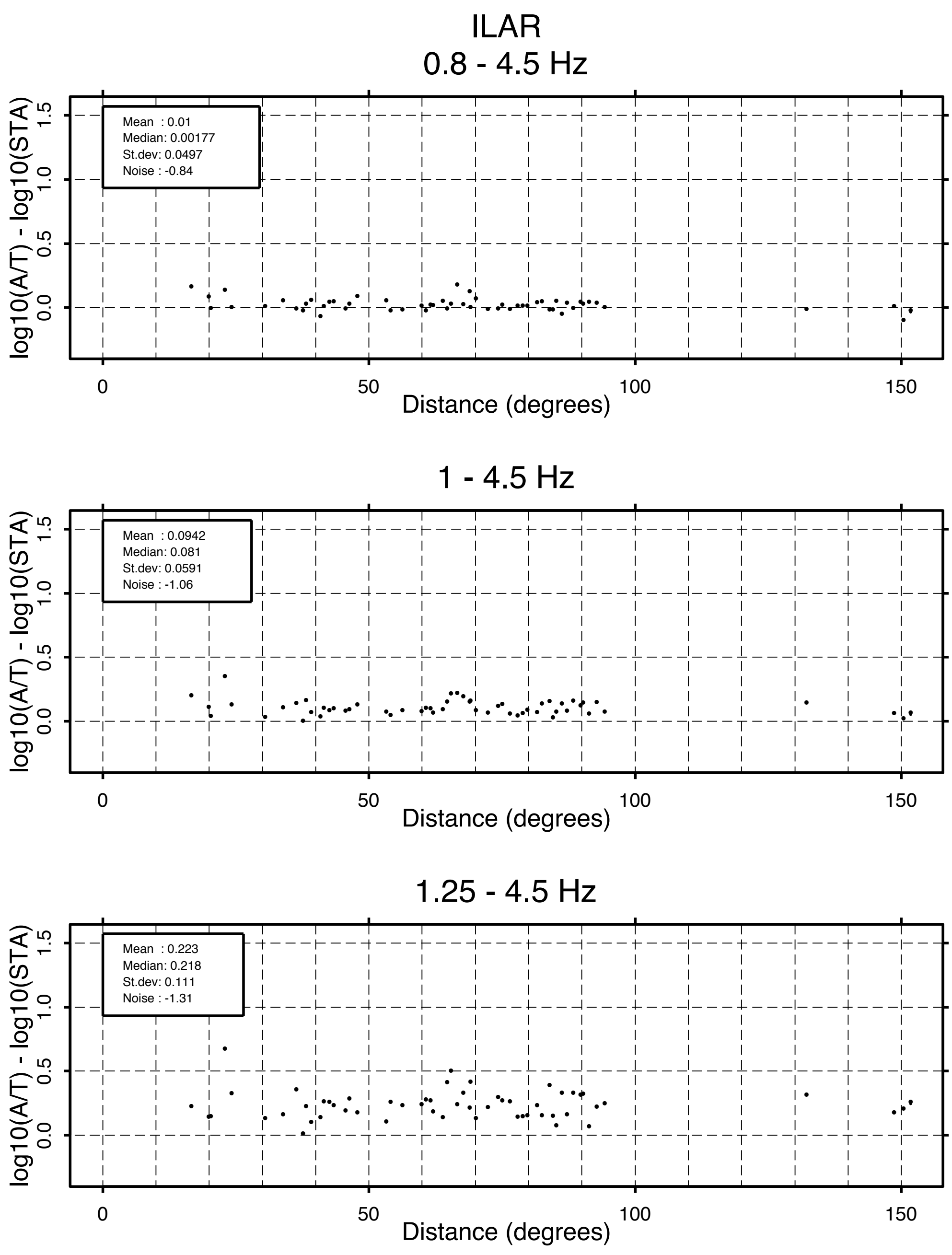


FIGURE 11. ATCOMP DATA FOR DISTANCES > 15 DEG.

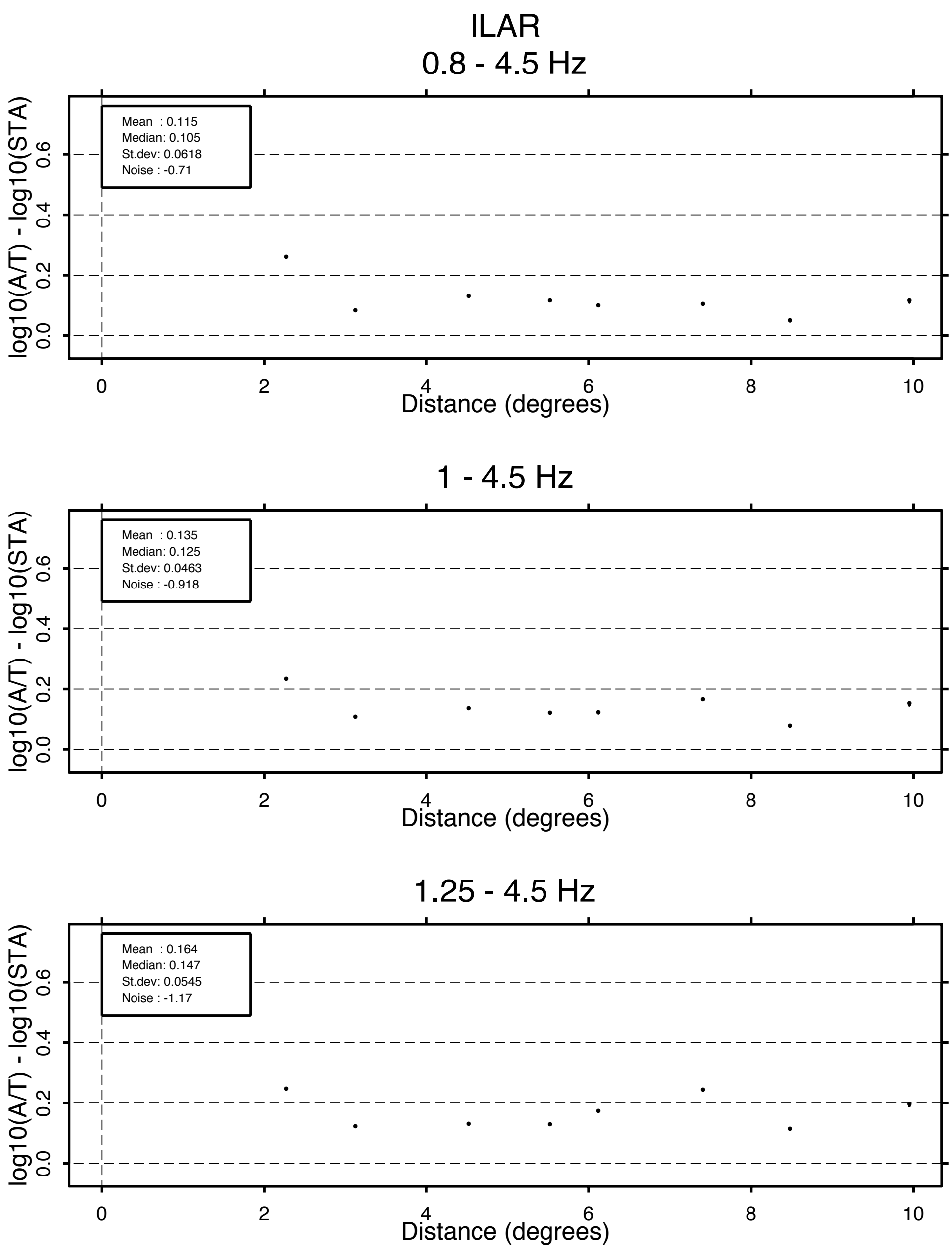


FIGURE 12. ATCOMP DATA FOR DISTANCES < 15 DEG.

Estimate Signal Loss and Mis-steering

This section addresses signal loss and beam mis-steering in seismic array data and how they affect detection threshold estimates in TM.

First, beamforming usually leads to a reduction in signal amplitude, and an estimate of this reduction must be available.

Second, when deploying a beam set for processing array data, the beams are usually deployed such that a maximum of 3 dB signal loss is allowed. Therefore, an estimate of the signal loss as a function of mis-steering is needed.

Third, each point of the global grid encompasses a circular region with a radius of about 2.7 degrees. When a given beam is used to monitor a given region, mis-steering of the beams must be taken into account to ensure conservative capability estimates.

For each event the signal loss as a function of the mis-steering of the beams has been estimated according to the relation:

sloss = (beam STA)/(Average STA of individual sensors)

where STA is taken to be the maximum within 8 seconds of the first arrival. Figure 13 shows the steering points used in the analysis with values relative to the observed azimuth and slowness of each event.

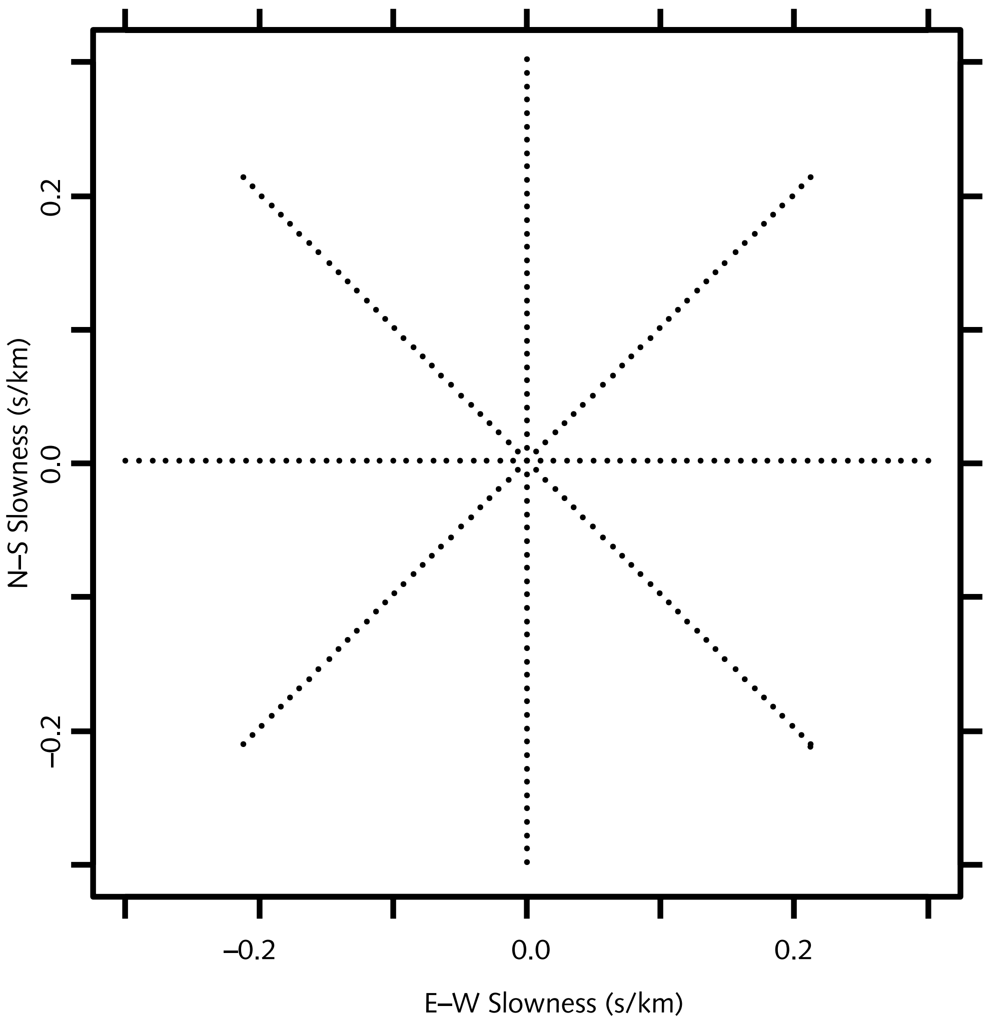


FIGURE 13. BEAMFORMING STEERING POINTS FOR ASSESSING BEAM MIS-STEERING SIGNAL LOSS

Figure 14 shows the signal loss as a function of mis-steering for ILAR events located at distances larger than 15 degrees. A 1.0–4.5 Hz bandpass prefilter has been applied to the data. The average signal loss for correct beam steering is 1.27 dB. An additional 3 dB signal loss is expected at a mis-steering of 0.038 s/km. The use of the 3 dB level for deploying a beam set is discussed in the next section. The corresponding curves for ILAR events located within 15 degrees are shown in Figure 15. The average beam loss for correct beam steering is 1.8 dB, and an additional 3 dB signal loss is expected at a mis-steering of 0.011 s/km. In this distance interval the 1.25–4.5 Hz bandpass prefilter has been applied.

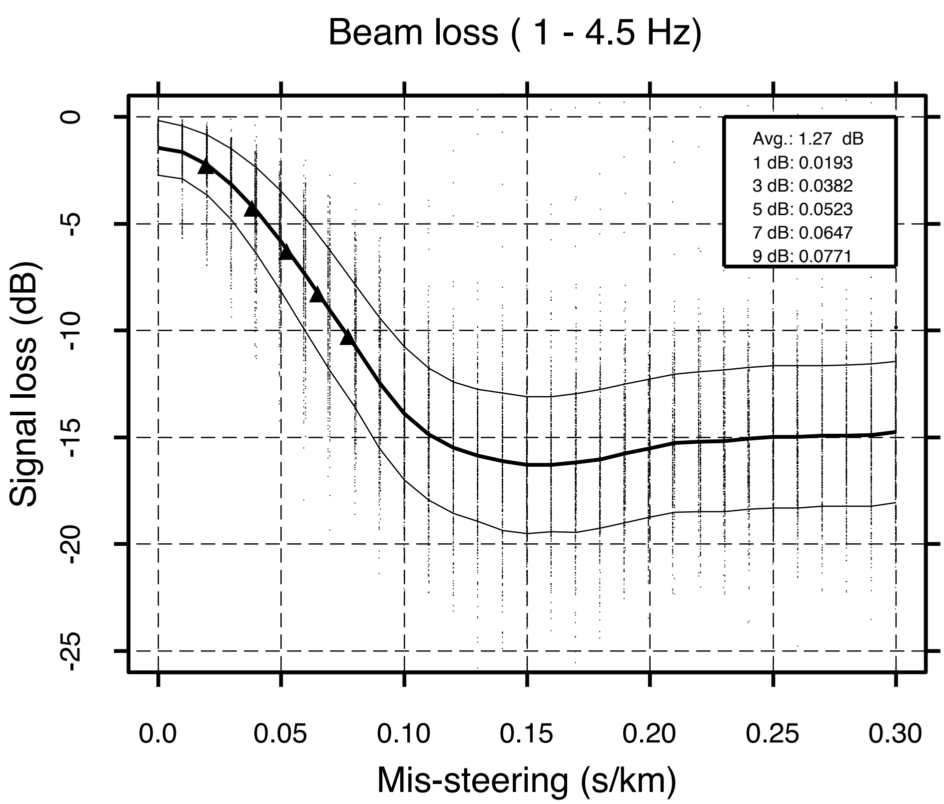


FIGURE 14. ILAR SIGNAL LOSS FOR EVENTS FURTHER THAN 15 DEG.

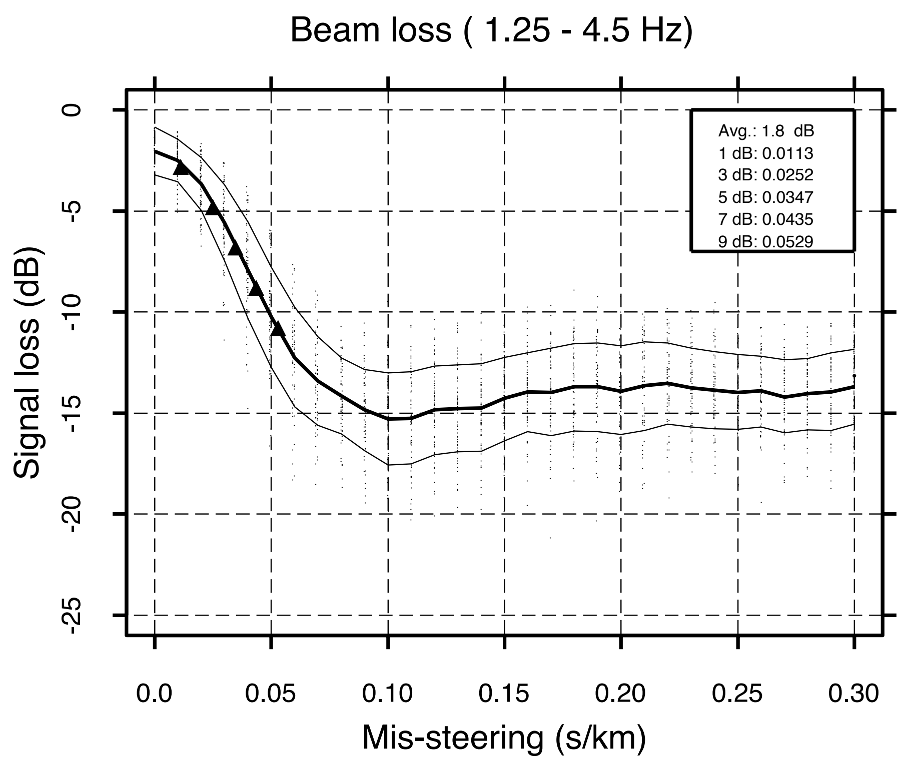


FIGURE 15. ILAR SIGNAL LOSS FOR EVENTS WITHIN 15 DEG.

Establish Beam Set

According to the IASP91 travel-time tables [Ken91b], the P and PKP phases used for estimating network detection thresholds span the slowness range 0.0–0.123 s/km for distances above 15 degrees. The ILAR beams in this distance range have an expected 3 dB signal loss at a mis-steering of 0.038 s/km (Figure 14).

The procedure for deploying the beams is illustrated in Figure 16 where circles with radii of 0.038 s/km (3 dB level) are fitted within the slowness range 0.0–0.123 s/km. The center points of the small circles correspond to the steering parameters of the beams.

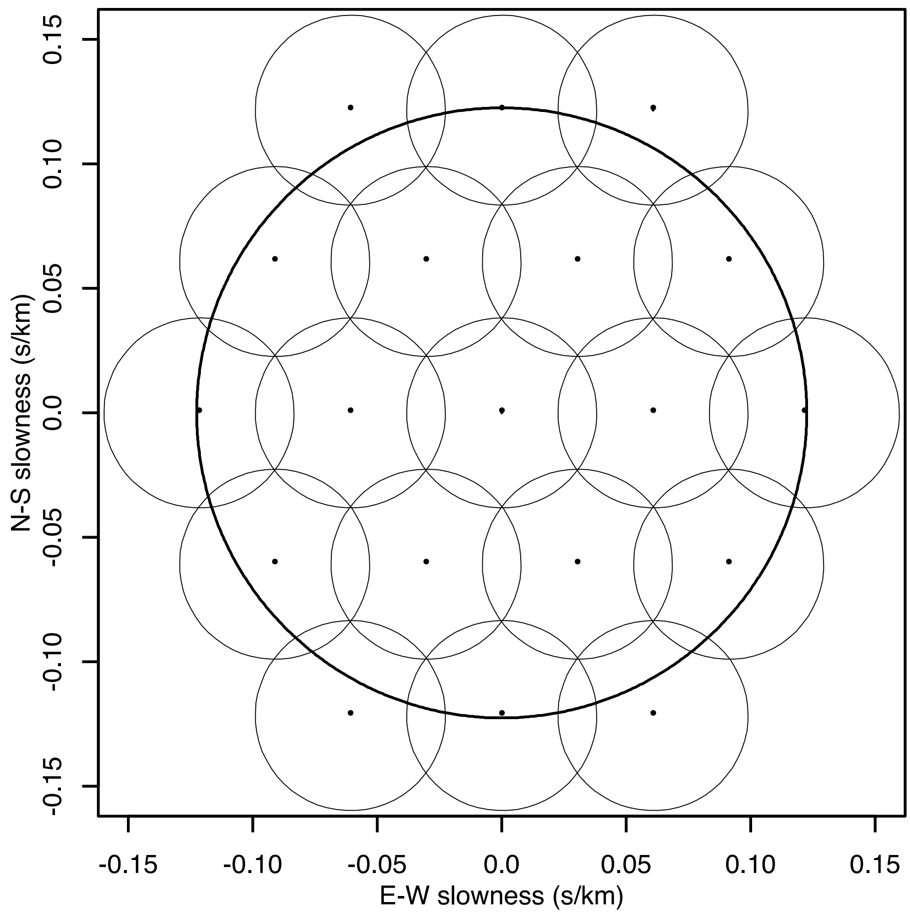


FIGURE 16. ILAR BEAM DEPLOYMENT FOR 15–180 DEG.

Excluding P phases from surface events at distances less than 1.4 degrees, there is very little difference between the theoretical slownesses in the 1.4 to 15 degree range (according to the IASP91 travel-time tables [Ken91b]). The slowness range is from 0.120 to 0.124 s/km, illustrated by the thin area between the two large circles of Figure 17. A set of 16 beams are needed to cover this slowness range with signal loss less than 3 dB. For the 1.25–4.5 Hz prefilter used to monitor events in this distance range, 3 dB signal loss is expected at a mis-steering of 0.011 s/km (see "Estimate Signal Loss and Mis-steering" on page 43). For distances within 1.4

degrees of ILAR this beam deployment has a signal loss somewhat greater than 3 dB. This implies that the estimates of the detection capability for this small region around ILAR are probably on the conservative side when using the beam deployment shown in Figure 17.

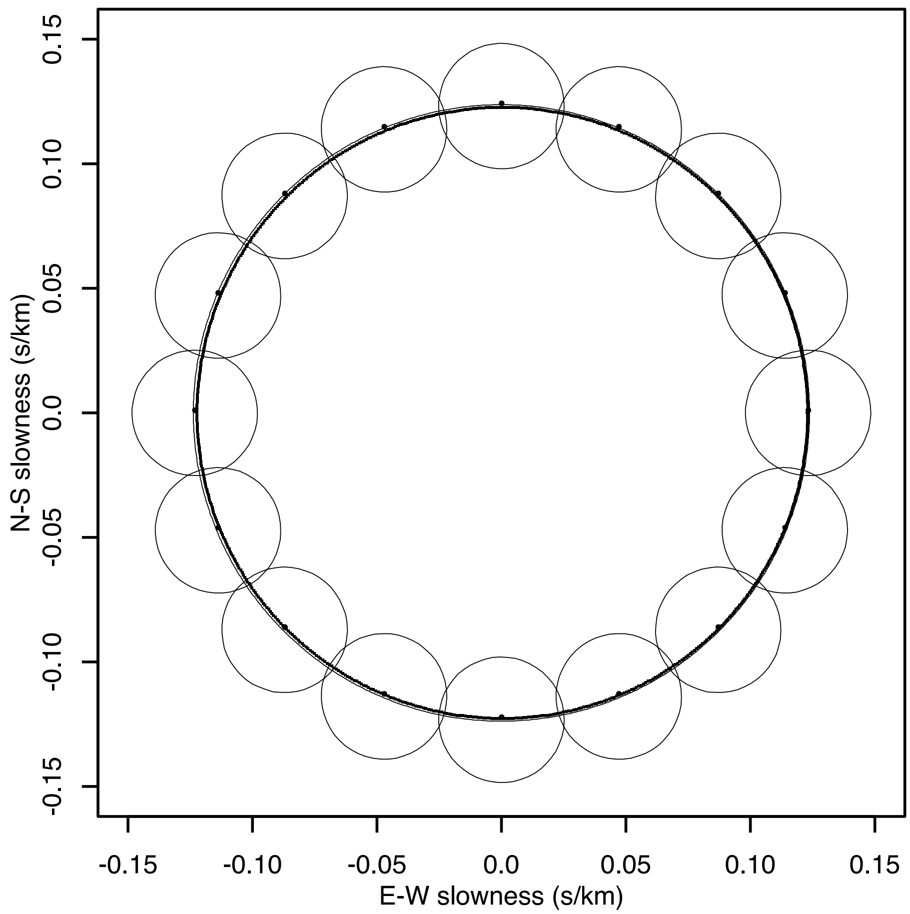


FIGURE 17. ILAR BEAM DEPLOYMENT FOR 1.4–15 DEG.

Chapter 3: Troubleshooting

This chapter describes how to identify and correct problems related to TM and includes the following topics:

- Monitoring
- Interpreting Error Messages
- Solving Common Problems
- Reporting Problems

Chapter 3: Troubleshooting

MONITORING

TM is monitored by examining log files. Log output from TM is directed to the following files:

TMthreshold	`$logdir/`*<yyyyddd>*`/TM/tmthreshold`
TMmap	`$logdir/`*<yyyyddd>*`/TM/TMmap`
TMprod	`$logdir/`*<yyyyddd>*`/TM/TMprod`
TMpostreb	`$logdir/`*<yyyyddd>*`/TM/TMpr`

Messages from *tuxshell* may also appear in these log files.

INTERPRETING ERROR MESSAGES

General Advice

If *CreateTMSession* is used to create the TM session, then all files in the session should be correctly formatted. This chapter does not describe error messages resulting from creating the session by hand, from running parts of the software (such as *tm_globrec* or *tm_beambasic*) standalone, or that occur as a consequence of changes to the software. It does describe some errors that result from problems with static data and errors that indicate a need to change hardcoded parameters in the software.

If a file cannot be opened, read, written, or closed check for the following causes:

- incompatibility between access codes for the file and privileges of the user
- existence of the file
- incorrect values for the system variables `$TM_HOME`, `$TM_SESSION`, `$TM_PRODDIR`, and `$PRI_LIST`
- typographical errors in system variables, parameter files, and so on
- external problems such as the lack of disk space

If TM fails because a file in the TM session is missing, unreadable, or corrupt, it may be easiest to use *CreateTMSession* to create a new session. There may be problems with other files, or parts of the directory structure may be missing. However, if most of the files are problem free and only a few have become corrupted or accidentally deleted, then you can create a new session with *CreateTMSession* and copy the desired files to the new session. You can also copy missing files from a new session to an old session created with the same parameters. In some cases, you can copy files from elsewhere (for example, `$TM_HOME/data/`) or write them manually.

Disk Loop Writing Codes

Disk loops are written in small data intervals by the subroutine *tmcfwrite.* If the verbosity level is at least 3, the subroutine prints messages that depend on the interval of data that was written to the output disk loop file (for *TMthreshold*, disk loop files are written to `$TM_SESSION/Region/TM/`<*tmtype*>`/Data/`). The message written is in the form "`Case` <*n*> `writing`" where the n case codes are defined in Table 2.

TABLE 2: DISK LOOP WRITING CODES

Case Code	Definition
1	Data were written to the disk loop for the first time.
2	Data to be written were all from a time earlier than the disk loop interval; no writing was done.
3	Data to be written overlapped the early end of the disk loop interval; only the data that were in the disk loop interval were written.
4	The entire disk loop interval was within the data interval to be written; the entire disk loop interval was overwritten, then data from times after the disk loop interval were written.
5	Data to be written were all within the disk loop interval; all data were written.
6	Data to be written overlapped the latter end of the disk loop interval; all data were written.
7	Data to be written were all from a time later than the end of the disk loop interval; the gap between the end of the disk loop interval time and the beginning time of the data was filled with NULLs, and all data were written.

If the amount of data to be written is longer than the maximum disk loop file length, then the earliest data are truncated.

General Error Messages

General error messages can occur in any of the TM FORTRAN programs (*TMthreshold*, *TMmap*, *tm_globrec*, *tm_beambasic*). A general error message is usually followed by a message from the calling program. The following list includes messages indicating that a parameter must be updated:

Message:	`FMBNAM: insufficient space for output name` or `Maximum length of filename=`*<lfnam>* or `FMICHK: Length of filename out of range=`*<lname>* `Range is 1 to` *<len>* `FMBNAM: error from FMBMDN=2` `FMICHK: Error from FMBNAM=<2 | 3>` `FMOPEN: return code from FMICHK=<7 | 8>` `FMOPEN: error on file:` *<filename>*
Description:	*FMBNAM* alters the input filename so that it conforms to the current system (leading and trailing blanks are removed, illegal characters are checked for, and so on). *FMICHK* passes a pointer to space for the new filename, which must be large enough.
Action:	Edit the parameter *LFNAM* (currently `220`) in `fman.h`. Then, recompile the library and all TM programs.

Message:	`FMBNAM: invalid input name, max length=500` `FMICHK: Error from FMBNAM=3` `FMOPEN: return code from FMICHK=7` `FMOPEN: error on file:` *<filename>*
Description:	*FMBNAM* alters the input filename so that it conforms to the current system (leading and trailing blanks are removed, illegal characters are checked for, and so on). In this case, the input filename is too long.
Action:	Edit the parameter *EFILI* (currently `500`) in `fmbnam.f`. Then, recompile the library and all TM programs.

Message: `FMICHK: Unit number out of range=`<*ifile*>
`Range is 0 to` <*nspace*>*-1*
`FMOPEN: return code from FMICHK=1`
`FMOPEN: error on file:` <*name*>

Description: The unit number is greater than the allowed number of units (currently `100`).

Action: Edit the range of unit numbers in `bdfman.f` (change the value of *HIGNUM*), and change the value of *NF1* in `fman.h`. Then recompile the library and all TM programs.

Message: `FMICHK: Invalid recordlength=`<*irecl*>
`Recordlength exceed maximum allowed by FMAN=`<*k*>

or

`Existing direct access file recordlength=`<*recl*>

or

`Existing sequential access file recordlength=`<*recl*>
`FMOPEN: return code from FMICHK=2`
`FMOPEN: error on file:` <*name*>

Description: The record length in bytes must be given to *FMOPEN*. If the record length is less than 1 or is too large, an error occurs.

Action: If the record length is too large, edit the maximum values in the subroutine *FMISDV* in file `fmunix.f`. The relevant parameters are *MRLSFF* for fixed length sequential formatted files, *MRLSFV* for variable length sequential formatted files, *MRLSUF* for fixed length sequential unformatted files, *MRLSUV* for variable length sequential unformatted files, *MRLDF* for direct access formatted files, and *MRLDU* for direct access unformatted files. The current value for *MRLSFV* and *MRLSUV* is `32756`; for the remaining parameters it is `32760`. Recompile the library and all TM programs after editing.

Message:	*<name>*`: Error from fmfrun`
Description:	No free FORTRAN unit number was available for opening files. The subroutines *gmtwrite*, *dintread*, and *dintwrite* can generate this error. These subroutines are found in `$CMS_HOME/src/automatic/src/TM/src/TMmap/` and `$CMS_HOME/src/automatic/src/TM/src/TMthreshold/`. *fmfrun* is a subroutine in the *ngut* library.
Action:	Edit the upper and lower limits for unit numbers (parameters *LOWNUM* and *HIGNUM*, currently `0` and `99`) in `bdfman.f` in the *ngut* library. Recompile the library and all TM programs.

TMthreshold

TMthreshold refuses to run if:

- *session_directory*, *t1*, *t2*, and *method* are not specified
- *ndetsta* (number of detecting stations) is less than one (default: `3`)
- *method* is neither `detection` nor `upplim` (upper or lower case)
- the format for *t1* or *t2* is invalid
- *t2* is not greater than *t1*

All subroutines of *TMthreshold* that are not in libraries are found along with the main program in the "program directory" (`$CMS_HOME/src/automatic/src/TM/src/TMthreshold/`). In some cases, copies of the same subroutine may be found in more than one program directory (for example, *gmtwrite* exists in both the *TMmap* and *TMthreshold* program directories).

Array Dimensions

Many variables have dimensions depending on parameters found in `$CMS_HOME/src/automatic/src/TM/src/TMthreshold/tmparm.h`.

- *mxglnumsta* (total number of stations allowed: `55`)

- *mxglnumphs* (total number of phases allowed: `2`)
- *mxglnumtyp* (total number of TM types allowed: `1`)
- *mxglnumtot* (total number of beams allowed for each target: `200`)
- *mxglnumtar* (total number of targets allowed: `2600`)

The software sends an error message and exits in most cases if these parameters must be increased, but if *mxglnumtot* is too small, it may crash.

The array dimension variables are saved in a common block and written to `$TM_SESSION/Region/TM/TM_recipes/compact` when *TMthreshold* runs. This file (instead of `trvminmax` and the target recipes in the same directory) is read (if available) when *TMthreshold* is executed again. If the information in this file is corrupted, you can delete it, and a new version based on the information in `trvminmax` and the target recipes will be written.

Time Intervals

Error messages from subroutines *tmstacfread*, *getavgmean*, *getavgmax*, and so on are usually due to a problem with the time interval. Extra checks on the time interval are included in the software, but these errors are unusual.

The time interval to be processed is normally divided into subintervals. The following internal variables are used to determine these subintervals (in subroutine *rprocess*):

- *deps* (seconds added to the end-time of the whole interval for comparison purposes; set in subroutine *rprocess*; default: `1e–3`)
- *trctrvmin* and *trctrvmax* (minimum and maximum travel times for the trace; read from recipes for targets in `$TM_SESSION/Region/TM/TM_recipes/`)
- *trvmin* and *trvmax* (minimum and maximum travel times for the station; read from `$TM_SESSION/Region/TM/TM_recipes/trvminmax`)
- *tmstep* and *stastep* (the same as the *tmsamp* and *stasamp* parameters from *CreateTMSession*, which are read from `$TM_SESSION/Region/process.parameters`)

Check that these values are reasonable. The time boundaries of the subinterval are never printed by the software. If *deps* must be changed, recompile *TMthreshold*. If any of the other values are unreasonable, change them to reasonable values and rerun *CreateTMSession*.

Checks done by the software before determining and processing subintervals should make these errors unlikely. If they do occur, and erroneous values for travel-times are found in the files listed above, then check for coordinate errors in `$TM_HOME/data/targets.`<*ntargets*> or travel-time table errors in `$TM_HOME/data/libdata/trtabl/`.

Error Messages

Message:	`Too many stations` `Increase array dimensions mxglnumsta`
Description:	The size of the station array (currently 55) is insufficient.
Action:	Edit the parameter *mxglnumsta* in `$CMS_HOME/src/automatic/src/TM/src/TMthreshold/tmparm.h`, and recompile the software.
Message:	`Too many phases` `Increase array dimensions mxglnumphs`
Description:	The size of the phase array (currently 2) is insufficient.
Action:	Edit the parameter *mxglnumphs* in `$CMS_HOME/src/automatic/src/TM/src/TMthreshold/tmparm.h`, and recompile the software.

Message:	`Too many TM types` `Increase array dimensions mxglnumtyp`
Description:	The size of the *tmtypes* array (currently `1`) is insufficient.
Action:	Edit the parameter *mxglnumtyp* in `$CMS_HOME/src/automatic/src/TM/src/TMthreshold/tmparm.h`, and recompile the software.

Message:	`Too many targets` `Increase array dimensions mxglnumtar`
Description:	Size of the target array (currently `2600`) is insufficient.
Action:	Edit *mxglnumtar* in `$CMS_HOME/src/automatic/src/TM/src/TMthreshold/tmparm.h`, and recompile *TMthreshold*.

Message:	`gmtwrite: Could not open or close` <*Session*>`/Region/TM/`<*tmtype*>`/Data/GMT.last`
Description:	The `GMT.last` file contains `0` when *TMthreshold* is running and `86400` when it is not. This error means that `GMT.last` could not be closed or opened for writing; it will most likely be preceded by one or more complaints from subroutines in the *ngut* library. `GMT.last` is written by *gmtwrite*, a subroutine of *TMthreshold*, and is not created by *CreateTMSession*. The *gmtwrite* subroutine is found in `$CMS_HOME/src/automatic/src/TM/src/TMthreshold/`.
Action:	The appropriate action depends upon the preceding error messages from routines in the *ngut* library. These messages often include (system dependent) *iostat* values.

Message:	`Error when opening` *<Session>*`/Region/TM/TM_recipes/stations.mask`
Description:	A file indicating that some stations should be masked exists, but is unreadable.
Action:	This file is not written by *CreateTMSession*. It contains a list (in one column) of station names to be masked. Recreate/fix it using a text editor. Data for stations listed in `stations.mask` are replaced by NULL values. Delete the file if no stations are masked.

Message:	`regprocess: Maxdat is too short`
Description:	The *maxdat* parameter is shorter than the maximum amount of data needed to span the time interval for the maximum number of traces. The error comes from *rprocess*, a subroutine in `$CMS_HOME/src/automatic/src/TM/src/TMthreshold/`.
Action:	The *maxdat* variable is set in `$CMS_HOME/src/automatic/src/TM/src/TMthreshold/chn000.h` to `800000`. This value is compared to a value calculated from parameters: *tmstep*, *stastep*, *trvmin*, and *trvmax*. The *tmstep* and *stastep* parameters are the same as *tmsamp* and *stasamp* written by *CreateTMSession* to `$TM_SESSION/Region/process.parameters`; *trvmin* and *trvmax* are in `$TM_SESSION/Region/TM/TM_recipes/trvminmax`. The calculated value is not printed. Try raising *MAXDAT*, and check that the four parameters listed above are reasonable. *TMthreshold* must be recompiled if *MAXDAT* is changed.

Message:	`getavgmean: ind1 is less than 1` or `getavgmax: ind1 is less than 1` or `getavgmean: ind2 is greater than nsamp` or `getavgmax: ind2 is greater than nsamp`
Description:	The index range for averaging in the current STA trace is less than 1 (FORTRAN indexing) or greater than the number of samples to be read. *getavgmean* and *getavgmax* are subroutines found in `$CMS_HOME/src/automatic/src/TM/src/TMthreshold/`.
Action:	The time intervals given to these subroutines should be reasonable. Check that the values discussed under "Time Intervals" on page 56 are reasonable; if not, change them accordingly. In addition, check that *tmlen* and *stalen* (read from `process.parameters`) are reasonable and that *tmlen* is greater than or equal to *stalen*. If not, correct the values, and recreate the directory structure with *CreateTMSession*.

TMmap

TMmap refuses to run if *session_directory*, *t1* (start-time), or *t2* (end-time) are not given. *session_directory* is the path and name of the session directory to be created; this should be the same as `$TM_SESSION`.

All subroutines of *TMmap* that are not in libraries are found along with the main program in `$CMS_HOME/src/automatic/src/TM/src/TMmap/`. In some cases, copies of the same subroutine may be found in more than one program directory (for example, *gmtwrite* exists in both the *TMmap* and *TMthreshold* program directories).

Error Messages

Message:	`FATAL ERROR: d2rproc: Processing interval not available in circular files`
Description:	The desired processing time interval does not fit within the available disk loop interval in `$TM_SESSION/Region/TM/`<*tmtype*>`/Data/`.
Action:	Check that the correct time interval was given.

Message:	`FATAL ERROR: d2rproc: Processing interval cannot be processed due to start of RDF file`
Description:	The desired start-time is less than the start-time in `$TM_SESSION/Region/TM/`<*tmtype*>`/RDF/data.interval`.
Action:	Check that the correct time interval was given.

Message:	`Error at 3` `FATAL ERROR: d2rtread: Too many targets!` or `Error at 5` `FATAL ERROR: tmtkcfread: Too many channels on stack`
Description:	Number of targets is equal to or greater than *MAXCHN* (currently `2600`).
Action:	Edit *MAXCHN* in `$CMS_HOME/src/automatic/src/TM/src/TMmap/d2r000.h`, and then recompile *TMmap*.

Message:	`Error at 5` `FATAL ERROR: Too much data for datarr`
Description:	This error is from *tmcfread*, a subroutine in the *tmcf* library. The number of samples requested, based upon the time interval requested and the sampling rate, is larger than the size of the array into which the data are read.
Action:	The size of this array is determined by parameter *MAXDAT* (currently `800000`). The size is reduced by the amount of data already read into the array. Change *MAXDAT* in `$CMS_HOME/src/automatic/src/TM/src/TMmap/d2r000.h`, and then recompile *TMmap*.
Message:	`RDFOPN: Could not open file!` `FATAL ERROR: d2rproc: Could not open rdf file` *<Session>*`/Region/TM/`*<tmtype>*`/RDF/grid.rdf`
Description:	The disk loop containing the TM map data could not be opened for updating. This message may be preceded by complaints from subroutine *FXOPEN* in library *ngut*.
Action:	Check permissions on the file. If there was a string allocation error (which implies that C was unable to assign space for the filename), then check for disk space or memory problems.
Message:	`OSCOM: Command too long. Longer than 499` `FATAL ERROR: d2rproc: Error from makecdf script`
Description:	The combined length of the name of the Bourne shell script *makecdf.2562*, *<scratch>*`/grid.xyz`, and *<scratch>*`/grid.cdf` is too long. The command consists of these three items in sequence.
Action:	Either change the scratch (or possibly the TM session) directory or recompile the *ngut* library and all TM programs after editing the parameter *LENBUF* (currently `500`) in `oscom.f`.

Message:	`FATAL ERROR: d2rproc: Conflict between cdf grid size and RDF grid size`
Description:	The RDF grid size does not exactly equal the Common Data Format (CDF) grid size (hardcoded to be 91 x 46 in the script *makecdf.2562*).
Action:	If a grid size other than 91 x 46 is desired, then a new version of the script *makecdf.2562* is required. Otherwise, recreate the session with *CreateTMSession* using *nxgrid* = `91` and *nygrid* = `46`.

TMprod and TMpostreb

The Bourne shell script *TMprod* will not run with fewer than six parameters. *TMpostreb* must have exactly four parameters to run.

These scripts do not generate specific error messages. If you find errors (for example, from *tuxshell*) in the log files, check the input parameters, system variables, configuration, and disk space.

CreateTMSession, AddTMStation, and DeleteTMStation

The *CreateTMSession*, *AddTMSession*, and *DeleteTMSession* C programs are used to initialize or alter the directory structure for threshold monitoring and are not executing during pipeline processing.

CreateTMSession, *AddTMStation*, and *DeleteTMStation* do not run if any of the following parameters are not specified:

- *par* (parameter filename)
- *session_directory* (path and name of session directory to be created; this should be the same as `$TM_SESSION`)
- *stlist* (station list)

CreateTMSession also refuses to run if the following parameters are missing:

- *staloop* (the size of the STA disk loops to be created)
- *tmloop* (the size of the TM disk loops to be created)

Error Messages

The error codes in C are implementation defined. Common error messages are described below:

Message:	`No such file or directory`
Action:	Check for typographical errors in the session directory, the static data directory, and the station list.
Message:	`No space left on device`
Action:	Check disk space. It may be necessary to delete other files or to choose another location for the TM session directory. Check that the parameters *staloop* and *tmloop* for *CreateTMSession* have been entered correctly.
Message:	`Permission denied`
Action:	Check the ownership of the directory in which the TM session resides or is to be written. Also check your current user ID. Remember that the TM session must be accessible by *DFX*.
Message:	`File exists`
Action:	*CreateTMSession* will not overwrite a previously existing session. Delete the old files and try again. *AddTMStation* will not overwrite existing stations; it skips them. Delete those to be rewritten with *DeleteTMStation* first.

Error messages from *CreateTMSession*, *AddTMStation*, and *DeleteTMStation* are often printed along with the C error messages such as those described previously.

Message:	`Problem opening` *<Home>*`/Data/Targets/targets.`*<ntargets>* `open:` *<C_error_message>* `Quitting.....`
Description:	*CreateTMSession* (using subroutine *copytf* in the *ctms* library) failed to open `$TM_HOME/data/Targets/targets.`*<ntargets>*. An error message from C should also be written.
Action:	This error may imply that the file does not exist. Check for typographical errors in `$TM_HOME`. Failing that, check that the file actually does exist; if it does not, it must be recreated (possibly by hand).

Message:	`tm_beambasic: Error from beambasic, Input file` *<Home>*`/data/Beamset/`*<stn>*`/beambasic` `(tm_beambasic completed)` `Problem opening` *<Session>*`/Region/STA/`*<stn>*`/DFXbeams` `open: No such file or directory` `Quitting.....`
Description:	`$TM_HOME/data/Beamset/` does not contain any files for *<stn>*. *tm_beambasic* is executed by *CreateTMSession* and *AddTMStation* (the FORTRAN code is found in `$CMS_HOME/src/automatic/src/TM/src/tm_beambasic`).
Action:	Check that *<stn>* was not misspelled and make sure that *<stn>* has a subdirectory that contains appropriate beam files in `$TM_HOME/data/Beamset/`.

Message: `Problem creating` <*Session*>`/Region/targets`
`creat:` <*C_error_message*>
`Quitting.....`

Description: *CreateTMSession* (using subroutine *copytf* in the *ctms* library) failed to open (to create) `$TM_SESSION/Region/targets`. An error message from C should also be written.

Action: Take action depending on the C error message. Lack of disk space is one likely error.

Message: `Write error on` <*file*>
`write:` <*C_error_message*>
`Quitting.....`

Description: The number of bytes read from `$TM_HOME/data/Targets/targets.`<*ntargets*> did not equal the number of bytes written to `$TM_SESSION/Region/targets`.

Action: Check disk space, or take other action as indicated by the C error message.

Message: `***> cannot open 1` <*disk_loop*>

or

`***> cannot write 2` <*disk_loop*>

or

`***> cannot close 3` <*disk_loop*>

Description: The *tmcfcreate* subroutine was unable to open, write, or close a disk loop for writing. Subroutine *tmcfcreate* is in the *tmcf* library.

Action: Check that the disk space is adequate. The disk loops are not the first files written, so write permissions should not be the problem.

Message:	`Input file` <*file*> `does not exist`
Description:	Either *beambasic* or *addread* was unable to open a file in `$TM_HOMEdata/Beamset/`<*stn*>. Both subroutines are located in `$CMS_HOME/src/automatic/src/TM/src/tm_beambasic/`.
Action:	Check for typographical errors in the path; then check that the files in question exist and are accessible.

Message:	`Cannot open` <*file*>
Description:	`TMbeams` and `DFXbeams` (in `$TM_SESSION/Region/STA/`<*stn*>) are opened for writing; other files are opened for reading. This error is from *beambasic*, *addread*, or *beamout*, which are subroutines found in `$CMS_HOME/src/automatic/src/TM/src/tm_beambasic/`.
Action:	If there is a problem writing a file, check disk space. Otherwise, check for typographical errors in `$TM_HOME`, and check that the named file exists.

Message:	`Format error in beam file information` <*beam_line*>
Description:	The number of items on a line read from `$TM_HOME/data/Beamset/`<*stn*>`/beambasic` is greater than 13 or less than 11. This error is from *beambasic*, a subroutine found in `$CMS_HOME/src/automatic/src/TM/src/tm_beambasic/`.
Action:	Check the format of the input file; it may need to be rewritten.

Message: `Unknown beam type` *<beamtype>*

Description: The "beam type" found in column two of `$TM_HOME/data/Beamset/`*<stn>*`/beambasic` is not `S` (for single channels) or `B` (for beams). This error is from *beamout*, a subroutine found in `$CMS_HOME/src/automatic/src/TM/src/tm_beambasic/`.

Action: Check the input file, and edit it if necessary. `B` and `S` are the only beam types allowed.

Message: `No new stations to add. Quitting.`

Description: This error is from *AddTMStation*. All stations in the station list already exist in `$TM_SESSION/Region/selected.stations`.

Action: Check that a new station list was given to the program. If a parameter file containing the original station list for *CreateTMSession* was used, a new station list must be added after the old parameter file on the command line.

Message: *<Session>*`:` *<C_error_message>*

Description: This error can occur when *AddTMStation* attempts to unlink `$TM_SESSION/Region/TM/TM_recipes/compact`.

Action: Check the file permissions. This file is written by *TMthreshold*. If there are no other problems you can delete it. *TMthreshold* will write a new version.

Other errors may occur when system commands are executed from within *CreateTMSession*, *AddTMStation*, or *DeleteTMStation*. The error messages that follow are from *tm_globrec* and its subroutines. *tm_globrec* is a FORTRAN program that is executed by *CreateTMSession*, *AddTMStation*, and *DeleteTMStation* via a system command.

The same parameters as those described for *TMthreshold* govern the array sizes used in *tm_globrec*. These parameters are found in `$CMS_HOME/src/automatic/src/TM/src/tm_globrec/tm_dirstruct.h`. The only difference between the parameters needed for *tm_globrec* and *TMthreshold* is that the value for *mxglnumsta* is `100` instead of `55`.

Message: `Site file` <*home*>`/data/libdata/TM.site does not exist`

Description: `$TM_HOME/data/libdata/TM.site` is a read-only ASCII dump of the **site** table [IDC5.1.1Rev2] containing coordinate information for the network. This error is from *globrecsta*, a subroutine in `$CMS_HOME/src/automatic/src/TM/src/tm_globrec/`.

Action: The `tm_globrec` file must be available if the TM processing recipes are to be written. Check that `$TM_HOME` points to the correct directory. If the file is missing, create it by dumping the table.

Message: `Error when opening TM.site`

Description: `$TM_HOME/data/libdata/TM.site` is a read-only ASCII dump of the **site** table [IDC5.1.1Rev2], containing coordinate information for the network. An error occurs in *globrecsta*, a subroutine in `$CMS_HOME/src/automatic/src/TM/src/tm_globrec/` when the file exists but cannot be read.

Action: Check and correct the read permissions for the file.

Message: `Too many beams`
`Increase array dimensions grmaxbeam`

Description: The size of the beam array (currently `50`) is inadequate.

Action: Edit *grmaxbeam* in `$CMS_HOME/src/automatic/src/TM/src/tm_globrec/tmgenbeams.h`.

Message:	`Have to set environment variable TM_HOME`
Description:	`$TM_HOME` is undefined. This system variable must be defined whether or not the *staticdir* argument is used with *CreateTMSession*, *AddTMStation*, or *DeleteTMStation*, because no parameters are passed directly to *tm_globrec*.
Action:	Define *TM_HOME*, for example, as follows: `% setenv TM_HOME=\` `/cmss/r2/config/app_config/monitoring/TM`
Message:	`***> goprsa: Increase dimension of maxwav` `tm_globrec: Error from globrec` `***> globchar: Error from goprsa` or `***> gltolico: Error from goprsa`
Description:	The maximum number of waves per station (currently `20`) is insufficient. *goprsa* is in the *go* library. *globchar, gltolico,* and *globrec* are subroutines of *tm_globrec* in `$CMS_HOME/src/automatic/src/TM/src/tm_globrec/`.
Action:	Edit the parameter *maxwav* in `goparm.h` (found with the *go* library source code) and then recompile the *go* library and all TM programs.

Message: `***>goprsa: You need to have the model file`
<staticdir>`/libdata/trtabl/tab.`*<phase>*
`tm_globrec: Error from globrec`
`***> globchar: Error from goprsa`

or

`***> gltolico: Error from goprsa`

Description: The travel-time table does not exist. *goprsa* is in the *go* library, and *globchar*, *gltolico*, and *globrec* are subroutines of *tm_globrec* in `$CMS_HOME/src/automatic/src/TM/src/tm_globrec/`.

Action: Check that `$TM_HOME` points to the correct directory. For each phase in `$TM_SESSION/Region/selected.phases`, make sure that a `tab.`*<phase>* file exists in `$TM_HOME/data/libdata/trtabl/`.

Message: `***> File `*<Home>*`/data/libdata/magtabl/tab.PTM`
`does not exist`
`tm_globrec: Error from globrec`
`globchar: Error from glmagtm`

Description: The attenuation data file is missing. *glmagtm* is in the *mag* library, and *globchar* and *globrec* are subroutines of *tm_globrec* in `$CMS_HOME/src/automatic/src/TM/src/tm_globrec/`.

Action: Check that `$TM_HOME` points to the correct directory. For each phase in `$TM_SESSION/Region/selected.phases`, make sure that a `tab.`*<phase>* file exists in `$TM_HOME/data/libdata/trtabl/`.

Message:	`? File` *<Home>*`/data/libdata/magtabl/tab.PTM` `will not open` `tm_globrec: Error from globrec` `globchar: Error from glmagtm`
Description:	The table containing attenuation data exists, but cannot be opened. The *glmagtm* subroutine is in the *mag* library. The *globchar* and *globrec* subroutines are part of *tm_globrec* in `$CMS_HOME/src/automatic/src/TM/src/tm_globrec/`.
Action:	Check and correct read permissions for `tab.PTM`.
Message:	`? Unexpected end-of-file on file` *<Home>*`/data/libdata/magtabl/tab.PTM` `tm_globrec: Error from globrec` `globchar: Error from glmagtm`
Description:	The attenuation table is incomplete. *glmagtm* is in the *mag* library. *globchar* and *globrec* are subroutines of *tm_globrec* in `$CMS_HOME/src/automatic/src/TM/src/tm_globrec/`.
Action:	Recreate the `tab.PTM` file.

SOLVING COMMON PROBLEMS

If the TM software stops operating, but does not provide an error message, check that the files in the static data directory (`$TM_HOME/data/`) do not contain errors. After checking the filesystems, restart the software.

REPORTING PROBLEMS

The following procedures are recommended for reporting problems with the application software:

1. Diagnose the problem as far as possible.
2. Record information regarding symptoms and conditions at the time of the software failure.
3. Retain copies of relevant sections of application log files.
4. Contact the provider or maintainer of the software for problem resolution if local changes of the environment or configuration are not sufficient.

Chapter 4: Installation Procedures

This chapter provides instructions for installing the software and includes the following topics:

- Preparation
- Executable Files
- Configuration Data Files
- Database
- Tuxedo Files
- Initiating Operations
- Validating Installation

Chapter 4: Installation Procedures

PREPARATION

Obtaining Released Software

The software is obtained via File Transfer Protocol (FTP) from a remote site or via a physical medium, such as tape or Compact Disk Read Only Memory (CD-ROM). The software and associated configuration data files are stored as one or more tape archive (tar) files. The software and data files are first transferred via FTP or copied from the physical medium to an appropriate location on a local hard disk. The tar files are then untarred into a standard directory structure.

Hardware Mapping

Software components are generally mapped to hardware to be roughly consistent with the software configuration model. Due to the amount of I/O performed, it is important that TM write to a locally mounted disk; otherwise, the performance of TM will be significantly degraded.

EXECUTABLE FILES

Use the script *makelibs* to make the libraries needed by the TM software and *makemods* to compile and link the TM software source code as follows:

```
% cd $CMS_HOME/src/automatic/src/TM
% makelibs
% makemods
```

After compilation, libraries specific to TM are placed in `$CMS_HOME/src/automatic/src/TM/lib/`. External libraries needed for linking with the TM software must be in `$CMS_HOME/src/`. The TM executable files are placed in `$CMS_HOME/bin/`.

CONFIGURATION DATA FILES

Place the configuration data files for TM in the proper directories, and then modify them to specify the details of the site configuration. The following sections describe the environment variables and configuration parameters.

Environment Variables

The directory paths listed in this chapter are an example configuration from the PIDC. Your configuration may differ.

The environment variables and paths discussed in this section are generally defined in terms of `$CMS_CONFIG`. At the PIDC, the environment variable `$CMS_CONFIG` is defined in the `.cshrc` file as `/cmss/config`. A few paths, notably in the Tuxedo parameter files, are given in terms of `$MONITORING-DIR`, which is defined in the `shared.par` file.

Global TM Variables

Global environment variables are defined in `$CMS_CONFIG/system_specs/env/global.env` as follows:

`$TM_HOME`	configuration directory for TM software
`$CMS_HOME`	software release directory for the CTBT Monitoring System (CMS)

The relevant lines from `$CMS_CONFIG/system_specs/env/global.env` follow:

```
setenv TM_HOME  /cmss/config/app_config/monitoring/TM
setenv CMS_HOME /cmss/rel
```

Shared Variables

Several variables are defined in `$CMS_CONFIG/system_specs/shared.par`:

`$LOGDIR`	log file directory
`$MONITORING-DIR`	path used in the Tuxedo parameter files
`$TM_SESSION`	TM intermediate files location needed by all of the TM programs and *DFX*
`$TM_PRODDIR`	TM output files location needed by *TMprod*
`$PRI_LIST`	primary seismic station list needed by *CreateTMSession*

The relevant lines from `$CMS_CONFIG/system_specs/shared.par` follow:

```
LOGDIR=/logs
MONITORING-DIR=$(CMS_CONFIG)/app_config/monitoring
TM_SESSION=/tm/session
TM_PRODDIR=/tm/products
PRI_LIST="'ABKT','ARCES','ASAR','BDFB','BGCA','BJT',
'BOSA','BRAR','CMAR','CPUP','DBIC','ESDC','FINES',
'GERES','HIA','ILAR','KBZ','KSAR','LPAZ','MAW','MJAR',
'NOA','NRIS','NVAR','PDAR','PDYAR','PLCA','ROSC','SCHQ',
'STKA','TXAR','ULM','VNDA','WRA','YKA','ZAL'"
```

Database Variables

Variables pertaining to the databases required during TM processing are defined in `$CMS_CONFIG/system_specs/process.par`:

`$IDCXDB` file products database needed by *TMprod* and *TMpostreb*

`$REBDB` REB database needed by *TMpostreb*

`$CMS_CONFIG/system_specs/process.par` is protected from reading by unauthorized users as this file contains database passwords. The format of the variables follows:

REBDB=*account*/*password*@*database_instance*
IDCXDB=*account*/*password*@*database_instance*
par=$(CMS_CONFIG)/system_specs/shared.par

DFX

DFX must run with TM parameters for each station used for threshold monitoring. *DFX* must be told where to find the TM parameters, what beams will be processed, and whether or not TM processing is to be performed.

TM-specific parameters are passed to *DFX* as regular and tabular parameter files using the *getpar* command line parser [Wah96]. These files are described in the sections that follow. Other TM-related items, such as `$CMS_CONFIG/app_config/DFX/scheme/DFX-detection.scm`, should not require changes unless the entire detection processing system is to be reconfigured.

Only those files that may require changes for TM are described. Refer to the *DFX* Software User Manual for general information on the configuration of *DFX*.

Detection Parameters

The parameters described in Table 3 must be set in `$CMS_CONFIG/app_config/DFX/ DFX-site-detection.par`.

TABLE 3: DFX DETECTION PARAMETERS

Parameter	Description
perform-tm-processing	1 to perform TM processing, 0 to skip it
dfx-arraytm-beam-list	comma-separated default list of beams for regional arrays
dfx-sstm-beam-list	comma-separated default list of beams for single stations
dfx-larraytm-beam-list	comma-separated default list of beams for large arrays
dfx-tm-recipe	name of par file containing TM processing parameters
tm-beam-list	construction of beam list parameters according to network type
par	path and name of TM par file (`dfx-tm-recipe`)

An example of TM parameters in the *DFX* par file follows:

```
# True to perform tm processing
perform-tm-processing=1

# Application the tm beam list is network-type specific

dfx-arraytm-beam-list=TM001,TM002,TM003,TM004,TM005,
TM006,TM007,TM008,TM009,TM010,TM011,TM012,TM013,TM014,
TM015

dfx-sstm-beam-list=TM001

dfx-larraytm-beam-list=TM001,TM002,TM003,TM004,TM005,
TM006,TM007,TM008,TM009,TM010,TM011,TM012,TM013,TM014,
TM015

# Create default parameter file names
dfx-tm-recipe=tm.par

# Set tm-beam-list according to NetType
tm-beam-list=$(dfx-$(NetType)tm-beam-list)

# Load parameters
par=$(dfx-par-dir)/tm/$(dfx-tm-recipe)
```

Recipe Parameters

The `dfx-tm-recipe` file (`$CMS_CONFIG/app_config/DFX/tm/tm.par`) contains the parameters described in Table 4.

TABLE 4: DFX TM RECIPE PARAMETERS

Parameter	Description
tm-output-directory	directory where the STA disk loops will be written
tm-null-val	NULL value used in STA disk loops
tm-stav-len	STA window length in seconds
tm-stav-step	increment for moving the STA window in seconds
tm-cf-length	disk loop length in seconds (if *DFX* will create disk loops)

An example of the *DFX* recipe parameter file for TM follows:

```
tm-output-directory=$(TM_SESSION)/Region/STA
tm-null-val=-2.0
tm-stav-len=1.0
tm-stav-step=1.0
tm-cf-length=604800.0
```

Beam Configuration

Each station <*stn*> has its own set of tables containing the beam definitions needed by *DFX*. One of these tables contains TM parameters. Each file should include a list of TM beams to be processed (*tm-beam-list*) and a table of beam parameters (*beam-recipe*):

`$CMS_CONFIG/app_config/DFX/beam/detection/`<*stn*>`-beam.par`

Each beam in *tm-beam-list* must have a *beam_recipe* entry. A description of the <*stn*>`-beam.par` parameters is given in Table 5. Beam recipe parameters are estimated using the tuning results described in "Tuning Processing Parameters" on page 36.

TM-specific STA beams have 2 or 3 in the *std* column of the *beam-recipe*. A value of 3 is reserved for the last of the beams and triggers an update of the `data.interval` file. The TM-specific STA beams defined in the *beam-recipe* must correspond exactly with the beam names given in *tm-beam-list*.

TABLE 5: BEAM DEFINITION PARAMETERS

Parameter	Description
tm-beam-list	comma-separated list of *TM* beam names (all beams in Table 3 on page 80 must be listed)
beam-recipe	table of beam recipes, one line for each beam listed in *tm-beam-list*
1 *name*	name of beam
2 *type*	type of beam (`inc` for incoherent, `coh` for coherent, `rms` for incoherent root mean square)
3 *rot*	beam rotation (`no` for no rotation, `rad` for radial, or `tang` for tangential; if `rad` or `tang`, the elements in the beam recipe must be orthogonal horizontal pairs)
4 *std*	standard "sbsnr" flag (0 for not a standard beam, 1 for standard beam, 2 for *TM* beam, 3 for last *TM* beam)
5 *azi*	beam steering azimuth (deg.; –1 for detection azimuth of detection-based beams or station-to-event azimuth of origin beams)
6 *slow*	beam steering slowness (s/km; –1 for detection slowness of detection-based beams or computed slowness of *phase* for origin beams)
7 *phase*	phase name
8 *flo*	low frequency filter corner (Hz)
9 *fhi*	high frequency filter corner (Hz)
10 *ford*	Butterworth filter order
11 *zp*	zero-phase filtering flag (0 for non zero-phase filtering, 1 for zero-phase filtering)
12 *ftype*	filter type (`BP` for bandpass filter, `LP` for low-pass filter below *fhi*, `HP` for high-pass filter above *flo*, `BR` for band-reject filter)
13 *group*	beam group

An example showing the relevant beam configuration parameters for ILAR follows:

```
tm-beam-list="TM001,TM002,TM003,TM004,TM005,TM006,TM007,TM008,TM009,TM010,TM011,TM012,
TM013,TM014,TM015,TM016,TM017,TM018,TM019,TM020,TM021,TM022,TM023,TM024,TM025,TM026,
TM027,TM028,TM029,TM030,TM031,TM032,TM033,TM034,TM035"

#!BeginTable beam-recipe
|name     |type|rot  |std|snr   |azi    |slow    |phase    |flo   |fhi   |ford|zp|ftype|group    |
 TM001     coh  no    2   -1.0    0.000 0.12315  P         1.25   4.5    3     0  BP    vertical
 TM002     coh  no    2   -1.0   22.500 0.12315  P         1.25   4.5    3     0  BP    vertical
 TM003     coh  no    2   -1.0   45.000 0.12315  P         1.25   4.5    3     0  BP    vertical
 TM004     coh  no    2   -1.0   67.500 0.12315  P         1.25   4.5    3     0  BP    vertical
 TM005     coh  no    2   -1.0   90.000 0.12315  P         1.25   4.5    3     0  BP    vertical
 TM006     coh  no    2   -1.0  112.500 0.12315  P         1.25   4.5    3     0  BP    vertical
 TM007     coh  no    2   -1.0  135.000 0.12315  P         1.25   4.5    3     0  BP    vertical
 TM008     coh  no    2   -1.0  157.500 0.12315  P         1.25   4.5    3     0  BP    vertical
 TM009     coh  no    2   -1.0  180.000 0.12315  P         1.25   4.5    3     0  BP    vertical
 TM010     coh  no    2   -1.0  202.500 0.12315  P         1.25   4.5    3     0  BP    vertical
 TM011     coh  no    2   -1.0  225.000 0.12315  P         1.25   4.5    3     0  BP    vertical
 TM012     coh  no    2   -1.0  247.500 0.12315  P         1.25   4.5    3     0  BP    vertical
 TM013     coh  no    2   -1.0  270.000 0.12315  P         1.25   4.5    3     0  BP    vertical
 TM014     coh  no    2   -1.0  292.500 0.12315  P         1.25   4.5    3     0  BP    vertical
 TM015     coh  no    2   -1.0  315.000 0.12315  P         1.25   4.5    3     0  BP    vertical
 TM016     coh  no    2   -1.0  337.500 0.12315  P         1.25   4.5    3     0  BP    vertical
 TM017     coh  no    2   -1.0  206.565 0.13590  P         1.0    4.5    3     0  BP    vertical
 TM018     coh  no    2   -1.0  180.000 0.12155  P         1.0    4.5    3     0  BP    vertical
 TM019     coh  no    2   -1.0  153.435 0.13590  P         1.0    4.5    3     0  BP    vertical
 TM020     coh  no    2   -1.0  236.310 0.10957  P         1.0    4.5    3     0  BP    vertical
 TM021     coh  no    2   -1.0  206.565 0.06795  P         1.0    4.5    3     0  BP    vertical
 TM022     coh  no    2   -1.0  153.435 0.06795  P         1.0    4.5    3     0  BP    vertical
 TM023     coh  no    2   -1.0  123.690 0.10957  P         1.0    4.5    3     0  BP    vertical
 TM024     coh  no    2   -1.0  270.000 0.12155  P         1.0    4.5    3     0  BP    vertical
 TM025     coh  no    2   -1.0  270.000 0.06078  P         1.0    4.5    3     0  BP    vertical
 TM026     coh  no    2   -1.0    0.000 0.00000  P         1.0    4.5    3     0  BP    vertical
 TM027     coh  no    2   -1.0   90.000 0.06078  P         1.0    4.5    3     0  BP    vertical
 TM028     coh  no    2   -1.0   90.000 0.12155  P         1.0    4.5    3     0  BP    vertical
 TM029     coh  no    2   -1.0  303.690 0.10957  P         1.0    4.5    3     0  BP    vertical
 TM030     coh  no    2   -1.0  333.435 0.06795  P         1.0    4.5    3     0  BP    vertical
 TM031     coh  no    2   -1.0   26.565 0.06795  P         1.0    4.5    3     0  BP    vertical
 TM032     coh  no    2   -1.0   56.310 0.10957  P         1.0    4.5    3     0  BP    vertical
 TM033     coh  no    2   -1.0  333.435 0.13590  P         1.0    4.5    3     0  BP    vertical
 TM034     coh  no    2   -1.0    0.000 0.12155  P         1.0    4.5    3     0  BP    vertical
 TM035     coh  no    3   -1.0   26.565 0.13590  P         1.0    4.5    3     0  BP    vertical
#!EndTable
```

The weighting for the stations and channels to be included in the beam is determined by the *group* parameter in the *beam-recipe* table. For ILAR, this refers to a table called `vertical`, which is also used during general *DFX* beamforming. Not all stations use the same table of weights for *TM* processing. The tables must be defined in the following par files:

`$CMS_CONFIG/app_config/DFX/beam/<stn>-beam.par`

Table 6 describes the parameters for setting beam weights.

TABLE 6: BEAM WEIGHT PARAMETERS

Parameter	Description
beam-group	table of beam groups
1 *group*	name of beam group, one line for each beam group
group	beam weight tables, one table for each group listed in *beam-group*
1 *sta*	name of station
2 *chan*	name of channel
3 *wgt*	assigned weight

For example, YKA has a separate set of weights for *TM*, which are found in a table called *grouptm*. To define this new table, *grouptm* is included in the *beam-group* table:

```
#!BeginTable beam-group
|group     |
 vertical
 ykb0
 horizontal
 grouptm
#!EndTable
```

The *grouptm* table is included after the *vertcal*, *ykb0*, and *horizontal* tables:

```
#!BeginTable grouptm
|sta        |chan         |wgt|
 YKB3        sz            1
 YKB4        sz            1
 YKB6        sz            1
 YKB7        sz            1
 YKR5        sz            1
 YKR6        sz            1
 YKR7        sz            1
 YKR8        sz            1
 YKR9        sz            1
#!EndTable
```

TMthreshold

To calculate network detection thresholds for each target point, *TMthreshold* requires extensive information on how to combine the continuous STA data from each of the primary seismic stations. These recipe files are created during the setup of the TM processing environment using *CreateTMSession* (see "Initiating Operations" on page 103). In addition to the processing recipes, *TMthreshold* picks up a few input parameters either from the command line or from a parameter file. Table 7 describes the *TMthreshold* parameters.

TABLE 7: TMTHRESHOLD PARAMETERS

Parameter	Default	Description
session_directory	none (required)	location of the *TM* processing environment
t1	none (required)	start-time of the processing segment
t2	none (required)	end-time of the processing segment

TABLE 7: TMTHRESHOLD PARAMETERS (CONTINUED)

Parameter	Default	Description
verbose	none (optional)	if present, diagnostics are printed on `stdout`
method	none (required)	processing method; either `detection` (calculate 90 percent detection threshold) or `upplim` (calculate 90 percent upper magnitude limit for non-detected events)
ndetsta	3	number of stations required for event detection (used only if *method*=`detection`)
detsnr	`4.0`	snr required for phase detection for all stations (used only if *method*=`detection`)

An example parameter file follows:

```
par=$(IMSPAR)
session_directory=$(TM_SESSION)
method=detection
ndetsta=3
detsnr=2.5
verbose
```

TMmap

TMmap makes use of the GMT functions *blockmedian* and *surface* to interpolate and resample the detection thresholds [Wes95]. The results are stored in a disk loop where the interpolated detection thresholds for each time sample (at 10 second intervals) are stored sequentially.

TMmap picks up the parameters in Table 8 from the command line or from a parameter file.

TABLE 8: TMMAP PARAMETERS

Parameter	Default	Description
session_directory	none (required)	location of the *TM* processing environment
t1	none (required)	start-time of the processing segment
t2	none (required)	end-time of the processing segment
tmtype	`Network`	name of station subset used for calculating detection thresholds (`Network` indicates that all stations are used in the processing)
scratchdir	*<Session>*`/scratch`	name of temporary file storage directory
gmtscript	`makecdf.2562`	name of Bourne shell script doing the actual interpolation and reformatting of the data
verbose	none (optional)	if present, diagnostic information are printed on `stdout`

The PIDC parameter file `TMmap.par` follows. The character string `$TM_SESSION` is defined in `$IMSPAR`.

```
par=$(IMSPAR)
session_directory=$(TM_SESSION)
tmtype=Network
gmtscript=makecdf.2562
verbose
```

Static Data

The static data directory is shown in Figure 7 on page 20. Descriptions of the files and the parameters that they contain are given in the sections that follow.

Beamset/

<*staticdir*>`/Beamset/` contains a subdirectory for each station in the network. Each subdirectory contains a file with beam information (`beambasic`) and one or more files with steering parameters relevant to the station.

`beambasic` has one heading row and one row for each beam. The `beambasic` parameters (presented as columns of a table with a header) are described in Table 9. Beam parameters are estimated using the tuning results described in "Tuning Processing Parameters" on page 36.

TABLE 9: BEAMBASIC PARAMETERS

Column and Parameter		Description
1	*Name*	name of current beam (for example, `TM001`)
2	*Type*	either `"B"` (beam) or `"S"` (single channel)
3	*Azi*	azimuth for this beam (deg.)
4	*Slow*	slowness for this beam (s/km)
5	*STAstp*	STA sampling rate (s)
6	*STAlen*	length of STA window (s)
7	*Dlow*	closest distance from station for this beam
8	*Dhig*	farthest distance from station for this beam
9	*DFXbeam*	actual beam (written by *DFX*), which may differ from *Name*
10	*Steerpar*	file containing steering parameters for this beam
11	*Ref*	name of reference station (usually the current station)
12	*Freq1*	low end of filter bandpass
13	*Freq2*	high end of filter bandpass

Example beambasic file for GERES:

```
Name  Type  Azi     Slow     STAstp  STAlen  Dlow  Dhig  DFXbeam
Steerpar          Ref  Freq1  Freq2

TM001  B    0.000 0.11085   1.0     1.0    0.0   20.0   TM001
missteer.0-20    GERES  1.25   4.5

TM002  B   60.000 0.11085   1.0     1.0    0.0   20.0   TM002
missteer.0-20    GERES  1.25   4.5

TM003  B  120.000 0.11085   1.0     1.0    0.0   20.0   TM003
missteer.0-20    GERES  1.25   4.5

TM004  B  180.000 0.11085   1.0     1.0    0.0   20.0   TM004
missteer.0-20    GERES  1.25   4.5

TM005  B  240.000 0.11085   1.0     1.0    0.0   20.0   TM005
missteer.0-20    GERES  1.25   4.5

TM006  B  300.000 0.11085   1.0     1.0    0.0   20.0   TM006
missteer.0-20    GERES  1.25   4.5

TM007  B    0.000 0.00000   1.0     1.0   20.0  180.0   TM007
missteer.20-180  GERES  0.8    3.0
```

Files listed in column 10 of beambasic have names with the form:

missteer.<*Dlow*>-<*Dhig*>

where <*Dlow*> and <*Dhig*> are found in columns seven and eight of beambasic. The contents of the missteer.<*Dlow*>-<*Dhig*> files are described in Table 10. Single channel beams cannot include mis-steering or signal loss.

TABLE 10: MISSTEER.DLOW-DHIG PARAMETERS

Parameter	Description
ATcomp	average difference between log(A/T) between 0.8 and 4.5 Hz and log((π/2)·STA·*calib*) in three different frequency bands for events more than 15 degrees from the station
sloss	average reduction of signal amplitude after beamforming due to signal decorrelation (dB)
column 1	beam mis-steering (s/km)
column 2	signal loss (dB)

Example rows for `GERES missteer.20–180` follow:

```
A/Tcomp (mag) : -0.0706
sloss(dB)     : 0.123
sloss         :
0  0
0.01      0.0633
0.02      0.132
0.03      0.253
0.04      0.418
0.05      0.626
0.06      0.878
0.07      1.17
0.08      1.49
0.09      1.86
0.1       2.28
0.125     3.46
0.15      4.79
0.175     6.24
0.2       7.73
0.225     9.14
0.25     10.5
0.275    11.6
0.3      12.8
```

So, for a beam mis-steering 0.275 s/km, the signal loss is 11.6 dB.

libdata/

<*staticdir*>`/libdata/` includes `TM.site`, a dump of the IDC **site** database table. `TM.site` contains coordinate information for the network and is a read-only ASCII file. The parameters of `TM.site` are described in Table 11.

TABLE 11: TM.SITE PARAMETERS

Column and Parameter	Description
1 *sta*	station identifier
2 *ondate*	Julian start date
3 *offdate*	Julian off date

TABLE 11: TM.SITE PARAMETERS (CONTINUED)

Column and Parameter	Description
4 *lat*	latitude
5 *lon*	longitude
6 *elev*	elevation
7 *staname*	station description
8 *statype*	station type: `ss` = single station, `ar` = array
9 *refsta*	reference station for array members
10 *dnorth*	offset from array reference (km)
11 *deast*	offset from array reference (km)
12 *lddate*	load date

Three lines from the `TM.site` file follow:

```
NVAR    1997260     -1    38.4295 -118.3037    2.0400 Mina Array, Nevada
ar    NV01      0.0000    0.0000 19990126 21:52:24

NV01    1997260     -1    38.4295 -118.3037    2.0400 Mina Array, Nevada
ss    NV01      0.0000    0.0000 19990126 21:52:24

NV02    1997260     -1    38.4372 -118.3055    2.0970 Mina Array, Nevada
ss    NV01      0.7150   -0.1100 19990126 21:52:24
```

Attenuation curves are tabulated in the ASCII file `tab.PTM`, which is located in the `magtabl/` subdirectory of *<staticdir>*`/libdata/`. An ASCII travel-time table, also called `tab.PTM`, is found in the `trtabl/` subdirectory of *<staticdir>*`/libdata/`. PTM is the only phase for which attenuation curves and travel-time tables are included in *<staticdir>*/. The format for the `tab.PTM` files is described in Table 12.

TABLE 12: TAB.PTM PARAMETERS

Parameter	Description
filename	table to use for teleseismic P waves (`n` for N/A)
ndepth	number of depth samples
depths	blank separated depths (km)
ndist	number of distance sample
distances	blank separated distances (deg.)
data(depth[1],distance[i])	Q or travel time/amplitude for depth 1, one row per distance
…	…
data(depth[ndepth],distance[i])	Q or travel time/amplitude for depth *ndepth*, one row per distance

An example `tab.PTM` file for attenuation follows:

```
n # P magnitude table used for TM (a combination of P, PP, PKPdf, PKPab
 2    # number of depth samples
  0.00  100.00
181    # number of distance samples
  0.00   1.00   2.00   3.00   4.00   5.00   6.00   7.00   8.00   9.00
…
170.00 171.00 172.00 173.00 174.00 175.00 176.00 177.00 178.00 179.00
180.00

# Q factor for mb at z = 0.00 (zero-peak amplitude is input)
  0.000
  1.000
…
  4.256
  4.256

# Q factor for mb at z = 100.00 (zero-peak amplitude is input)
  0.000
  1.000
…
  4.256
  4.256
```

par/

Three subroutines of *TMprod* (*tmstatus*, *detplot*, and *plotuptime*) use the similarly named subdirectories of <*staticdir*>`/par/`.

Background noise level plots (Figure 5 on page 12) are created by *tmstatus*. Each plot in the figure has a corresponding line in <*staticdir*>`/par/tmstatus/tmstatus.dat` that contains the parameters listed in Table 13.

TABLE 13: TMSTATUS.DAT PARAMETERS

Column and Parameter	Description
1 *stn*	This parameter is the station name (for example, `ARCES`).
2 *missteer*	This parameter is the station teleseismic P-wave magnitude correction due to beamforming signal loss and a log(STA) correction (*sloss*/20 + *ATcomp*). These numbers are based on tuning studies (see "Tuning Processing Parameters" on page 36). For example, 0.09 m_b units must be added to log(STA) from ILAR to make it comparable with log(A/T). In addition, the expected signal loss from beamforming is 1.27 dB for teleseismic events. The total correction for ILAR is 0.16 m_b units.
3 *stntype*	This parameter provides information on the number of beams used to monitor the teleseismic distance regime. When a single beam is used (as for 3-C stations and small arrays where a single beam covers the teleseismic distances with less than 3 dB signal loss) *stntype* = `single`. For larger arrays, several beams are needed to monitor the teleseismic distance regime, and *stntype* = `group`.
4 *beamfile*	The *beamfile* is the name of the STA trace(s) to be used to represent the noise levels in the teleseismic distance regime and is related to the beam definitions of *DFX*. For 3-C stations and small arrays (*stntype* = `single`), one STA trace calculated from the Z-component is used. For larger arrays where several beams are used in the teleseismic regime (*stntype* = `group`), *beamfile* is the name of the file containing the beam names. For example, ILAR traces contains STA beams TM017–TM035. All 19 STA beams are analyzed to provide the noise levels for the status plots.
5 *fband*	This parameter is the frequency band used to monitor the teleseismic distance regime (for plotting purposes only).

An example of a few lines from a `tmstatus.dat` file follows (maintaining this par file is made easier if stations are listed in alphabetical order):

```
ABKT    0.03 single  TM001        0.8-3.0Hz
ARCES  -0.08 group   ARCES.traces 1.5-6.0Hz
ASAR    0.12 group   ASAR.traces  1.0-4.5Hz
BDFB    0.11 single  TM001        1.0-4.5Hz
BGCA    0.19 single  TM001        1.25-4.5Hz
BJT    -1.69 single  TM001        0.8-3.0Hz
```

Files listed in column 4 of `tmstatus.dat` for group station types are named <*stn*>`.traces` (for example, `ARCES.traces`). These files contain the list of required beam files (for example, `TM001`), one per line, that are found in the <*Session*> directory structure (see "CreateTMSession Output" on page 106).

Two color palette files are used by *detplot*. These files, `data.cpt.2` and `data.cpt.3`, are stored in <*staticdir*>`/par/detplot/`. Figure 6 on page 13 is created by *detplot*.

TABLE 14: DATA.CPT PARAMETER FILES

Line		Description
1	*contour interval 1*	low contour value, RGB values (0–255) of low contour, high contour value, RGB values (0–255) of high contour
2–*n*	…	red value of low contour (0–255)
n	*contour interval n*	low contour value, RGB values (0–255) of low contour, high contour value, RGB values (0–255) of high contour
n+1	*background color*	"`B`" followed by RGB values (0–255) of background color
n+2	*foreground color*	"`F`" followed by RGB values (0–255) of foreground color

An example `data.cpt` file follows:

```
# V2.0 cpt file created by makecpt on Sep 21 14:37:40 1994
# No input V1.0 shade table was given; an iso-hue rainbow
# was made. Contours were made using a mid-value of 4 and
# a contour interval of 0.5. Color palette is continuous.
#
3.0    0    0  246    3.5    0  255  255
3.5    0  255  255    4.0  130  255    0
4.0  130  255    0    4.5  252  246   23
4.5  252  246   23    5.0  250  170   66
5.0  250  170   66    5.5  178   34   34
B    0    0    0
F  255  255  255
```

The subdirectory *<staticdir>*`/par/uptime/` (used by script *plotuptime*) contains two ASCII files, `cur_pri.data` and `offset.data`. Information about the current primary network is stored in `cur_pri.data` in four columns. The columns of the parameter table in `cur_pri.data` are described in Table 15.

TABLE 15: CUR_PRI.DATA PARAMETERS

Column and Parameter	Description
1 *stn*	station name (for example, `ARCES`)
2 *lat*	latitude (deg., for example, `69.5349`)
3 *lon*	longitude (deg., for example, `25.5058`)
4 *statype*	two-character string (`"ss"` for single stations or `"ar"` for arrays)

An example of the first few lines of the `cur_pri.data` file follows (maintaining this par file is made easier if stations are listed in alphabetical order):

```
ABKT      37.9304   58.1189 ss
ARCES     69.5349   25.5058 ar
ASAR     -23.6664  133.9044 ar
BDFB     -15.6440  -48.0141 ss
BGCA       5.1761   18.4242 ss
```

Label placement information for each station, which is used by GMT programs when creating Figure 4 on page 11, is contained in `offset.data`. Table 16 gives the parameters of `offset.data`.

Table 16: offset.data Parameters

Column and Parameter	Description
1 *stn*	station name (for example, `ARCES`)
2 *xoff*	x offset (deg.)
3 *yoff*	y offset (deg.)
4 *justify*	justification code (1 = lower left, 2 = lower center, 3 = lower right, 5 = mid left, 6 = mid center, 7 = mid right, 9 = upper left, 10 = upper center, 11 = upper right)

An example of the first few lines of an `offset.data` file follows (maintaining this par file is made easier if stations are listed in alphabetical order):

```
ABKT    -2    0   10
ARCES    2   -2    3
ASAR    -2   -2   11
```

Targets/

The file <*staticdir*>`/Targets/targets.`*n* (where *n* is the number of targets) is an ASCII file containing the parameters listed in Table 17. Currently there are 2,562 targets, all of which have depths of 0.00 km and radii of 2.74 degrees. Figure 18 shows a map of some of the targets in this grid.

TABLE 17: TARGETS.N PARAMETERS

Column and Parameter		Description
1	*name*	target name (`Target_`*seqnum*, for example, `Target_1001`)
2	*lat*	latitude (deg., for example, `26.72`)
3	*lon*	longitude (deg., for example, `–162.00`)
4	*depth*	depth (km, for example, `0.00`)
5	*radius*	radius of the target region (deg., for example, `2.74`)
6	*seqnum*	sequence number (beginning with `1001`)

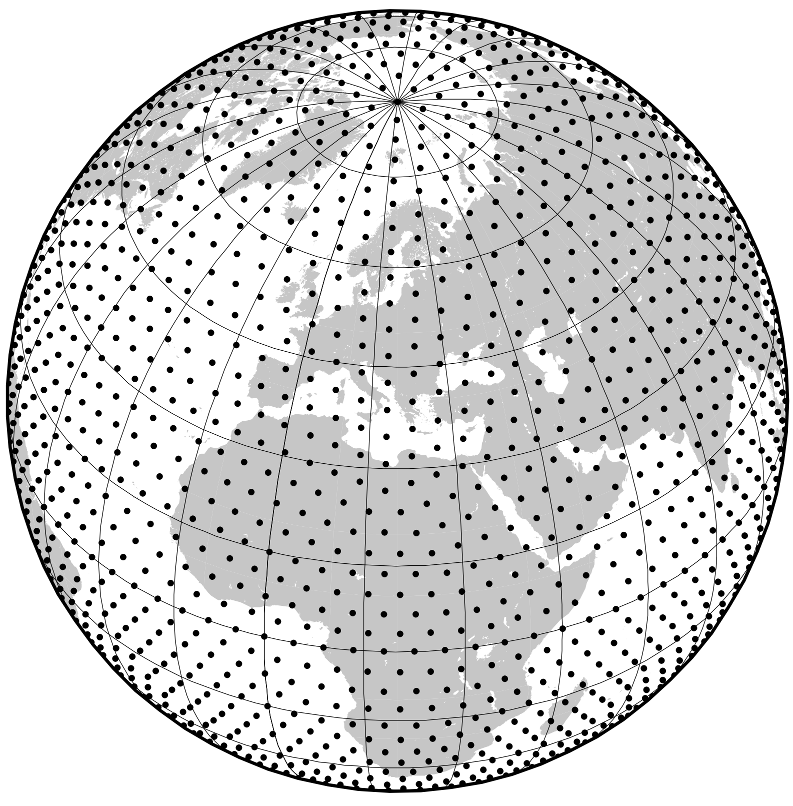


FIGURE 18. GLOBAL TARGET POINTS

An example of a few rows from `targets.2562` follows:

```
Target_1001    90.00     0.00    0.00    2.74    1001
Target_1002    26.72    90.00    0.00    2.74    1002
Target_1003    26.72   162.00    0.00    2.74    1003
```

DATABASE

This section describes database elements required for operation of this software component.

Accounts

The IDCX, SEL3, and REB database accounts [IDC5.1.3Rev0.1] are used and must be readable by TM.

Tables

TM requires the **fpdescription**, **fileproduct**, and **interval** database tables [IDC5.1.1Rev2]. The **fpdescription** table of the `$IDCXDB` account requires the following rows for TM:

TABLE 18: TM FPDESCRIPTION ROWS

typeid	prodtype	Name	msgd type	msgd format	header_ fpid	lddate
1	tm_uptime	Data Availability	bin	ps	–1	*dd-MMM-yy*
2	tm_status	Station Noise Levels	bin	ps	–1	*dd-MMM-yy*
3	tm_detection	Detection Capability	bin	ps	–1	*dd-MMM-yy*

An example of a load date (*lddate*) is `25-JUN-98`.

TUXEDO FILES

The `$TM_HOME` directory contains parameter files for the programs *CreateTMSession*, *TMthreshold*, and *TMmap*; the paths in the DACS parameter files must reflect the actual locations of `TMthreshold.par` and `TMmap.par`.

The DACS parameter files for the TM programs *TMthreshold*, *TMmap*, *TMprod*, and *TMpostreb* are found in `$CMS_CONFIG/app_config/distributed/tuxshell/tm/`. The parameter files include: `tuxshell-TMthresh.par`, `tuxshell-TMmap.par`, `tuxshell-TMprod.par`, and `tuxshell-TMpr.par`. The *tuxshell* parameters used in these files are described in Table 19. See [IDC6.5.2Rev0.1] for a detailed description of the DACS and the man page for *tuxshell* for a complete description of the parameters.

TABLE 19: TUXSHELL PARAMETERS

Parameter	Default	Description
role	none (required)	role name for *CommAgent* connection
prefix	*role*	replacement string for *role*
par	none	par file
sendkeys	1	flag that determines if the child process command line will contain key values (1 = yes)
prefix-exec	none (required)	*TMthreshold* executable
prefix-key[i]	none	string searched for in the data field of incoming messages
prefix[i]	none	arguments to child process
prefix-timeout	no time limit	time in seconds before terminating a running child
prefix-true-exit	0	if a child exit code matches a value in this array, then the child exit was a success
interval-source	none	source of interprocess communication (IPC) messages sent to *tuxshell* (`queue` or `tpcall`)

TABLE 19: TUXSHELL PARAMETERS (CONTINUED)

Parameter	Default	Description
interval-success-state	none	name of the interval or request state that indicates a successful run by the child
destqueue	none	destination queue (next pipeline step)
log	`stderr`	*tuxshell* log destination
outfile	`stdout`	standard output destination for child
errfile	`stderr`	standard error destination for child
logarguments	1	log arguments to *tuxshell* and child log files
TM-bulletin	none	file containing the REB used by *TM*

An example of `tuxshell-TMthresh.par` follows:

```
role=TMthreshold
prefix=TMthres  #cannot = role, because state cannot be
                #longer than 15 char: "TMthresh-started"

par=$(IMSPAR)
par=$(DISTRIBUTED)

sendkeys=0

TMthres-exec=$(RELBIN)/TMthreshold

TMthres-key[0]=time
TMthres-key[1]=endtime

TMthres[1]="par=$(MONITORING-DIR)/TM/TMthreshold.par"
TMthres[2]="t1=$(time)"
TMthres[3]="t2=$(endtime)"

TMthres-timeout=3600
TMthres-true-exit=0
```

```
# Queuing Flow
#
interval-source=queue
destqueue=TMmap

log=$(LOGDIR)/%jdate/tuxshell/$(role)-%host-%pid
outfile=$(LOGDIR)/%jdate/TM/$(role)
errfile=$(outfile)
logarguments=0
```

An example of `tuxshell-TMmap.par` follows:

```
role=TMmap
prefix=$(role)

par=$(IMSPAR)
par=$(DISTRIBUTED)

sendkeys=0

TMmap-exec=$(RELBIN)/TMmap

TMmap-key[0]=time
TMmap-key[1]=endtime

TMmap[1]="par=$(MONITORING-DIR)/TM/TMmap.par"
TMmap[2]="t1=$(time)"
TMmap[3]="t2=$(endtime)"

TMmap-timeout=1800
TMmap-true-exit=0

# Queuing Flow
#
interval-source=queue
destqueue=TMprod

log=$(LOGDIR)/%jdate/tuxshell/$(role)-%host-%pid
outfile=$(LOGDIR)/%jdate/TM/$(role)
errfile=$(outfile)
logarguments=0
```

An example of `tuxshell-TMprod.par` follows:

```
role=TMprod
prefix=$(role)

par=$(IMSPAR)
par=$(DISTRIBUTED)

sendkeys=0

TM-bulletin=TMbulletin.IDCREB

TMprod-exec=$(RELBIN)/TMprod

TMprod-key[0]=time
TMprod-key[1]=endtime

TMprod[1]="$(TM_SESSION)"
TMprod[2]="$(TM_PRODDIR)"
TMprod[3]="$(time)"
TMprod[4]="$(endtime)"
TMprod[5]="$(IDCXDB)"
TMprod[6]="$(TM-bulletin)"

TMprod-timeout=1800
TMprod-true-exit=0

# Queuing Flow
#
interval-source=queue
destqueue=done

interval-success-state="done"  # final SEL3 process

log=$(LOGDIR)/%jdate/tuxshell/$(role)-%host-%pid
outfile=$(LOGDIR)/%jdate/TM/$(role)
errfile=$(outfile)
logarguments=0
```

An example of `tuxshell-TMpr.par` follows:

```
role=tmpr
prefix=$(role)

par=$(IMSPAR)
par=$(DISTRIBUTED)

sendkeys=0

tmpr-exec=$(RELBIN)/TMpostreb

tmpr-key[0]=time
tmpr-key[1]=endtime

tmpr[1]="$(endtime)"
tmpr[2]="$(TM_SESSION)"
tmpr[3]="$(TM_PRODDIR)"
tmpr[4]="$(REBDB)"

tmpr-timeout=1800
tmpr-true-exit=0

# Queuing Flow
#
interval-source=queue
destqueue=done

interval-success-state=done

log=$(LOGDIR)/%jdate/tuxshell/$(role)-%host-%pid
outfile=$(LOGDIR)/%jdate/TM/$(role)
errfile=$(outfile)
logarguments=0
```

INITIATING OPERATIONS

Before TM processing can begin, a processing environment must be built. *CreateTMSession* creates this environment including the disk loop files, the various parameter and recipe files, and the directory structure. Instructions for using

CreateTMSession are provided in "Creating TM Sessions" on page 29. The following section describes the configuration of the *CreateTMSession* input files and output processing environment.

CreateTMSession Input

CreateTMSession has a number of arguments. Although it is possible to enter every argument on the command line it is most convenient to use a parameter file. The par file may contain the parameters shown in Table 20.

TABLE 20: CREATETMSESSION PARAMETERS

Parameter	Default	Description
session_directory	none (required)	fully qualified path of session directory <*Session*> to be created (normally `$TM_SESSION` for pipeline processing)
tmtype	`Network`	currently, *tmtype* should always be `Network`
phases	`PTM`	comma-separated list of phases to process (for example, *phase1*, *phase2*, …, *phasen*)
stlist	none (required)	comma-separated list of stations to use (for example, *stn1*, *stn2*, …, *stnn*)
staticdir	`$TM_HOME/data`	fully qualified path of static data directory *staticdir*
ntargets	`2562`	number of targets to use (considered a string by the program and used to determine the targets file to copy)
staloop	none (required)	duration, in seconds, of the STA disk loops (for example, `TM001`)
stasamp	`1.0`	time, in seconds, between the centers of successive STA windows
stalen	`1.0`	length, in seconds, of the STA window
stanullval	`–2.0`	value to be used in STA data files where there are no data

TABLE 20: CREATETMSESSION PARAMETERS (CONTINUED)

Parameter	Default	Description
tmloop	none (required)	length, in seconds, of the *TM* disk loops (for example, `Target_1001`)
tmsamp	`10.0`	distance, in seconds, between the centers of successive *TM* windows
tmlen	`1.0`	length, in seconds, of the *TM* window
tmnullval	`–2.0`	value to be used in *TM* data files where there are no data
nxgrid	`91`	size, in pixels, of the `grid.rdf` file (*x* axis)
nygrid	`46`	size, in pixels, of the `grid.rdf` file (*y* axis)
magstdev	`0.38`	standard deviation assumed for the P magnitude estimates
magconf	`0.90`	confidence level of upper magnitudes limits
search1	`6.0`	length of search interval after predicted P wave onset
search2	`3.0`	additional search interval due to travel-time uncertainties

The *staticdir* parameter is used to access several files that are used by *CreateTMSession*. These files include the following:

<*staticdir*>`/Targets/targets.2562`
<*staticdir*>`/Beamset/*`
<*staticdir*>`/libdata/TM.site`
<*staticdir*>`/libdata/magtabl/tab.PTM`
<*staticdir*>`/libdata/trtabl/tab.PTM`

The directory structure of these files is shown in Figure 7 on page 20, and the contents of the files are described in "Static Data" on page 87.

An example of a *CreateTMSession* parameter file follows. As long as there are no blank spaces, it is not necessary to enclose the strings in quotes. Do not leave spaces on either side of the equals signs (=). In this example, the environment variable `$TM_SESSION` is defined in the parameter file `$IMSPAR`. The session directory should equal `$TM_SESSION` for pipeline processing.

```
par=$(IMSPAR)
session_directory=$(TM_SESSION)
phases=PTM
stlist=$(PRI_LIST)
magstdev=0.38
magconf=0.90
search1=6.0
search2=3.0
tmsamp=10.0
tmlen=1.0
stasamp=1.0
stalen=1.0
ntargets=2562
staloop=604800.0
stanullval=-2.0
tmloop=604800.0
tmnullval=-2.0
nxgrid=91
nygrid=46
```

CreateTMSession Output

CreateTMSession generates a directory structure for TM processing. The directory structure, <*Session*>, is shown in Figure 19 and is described in the sections that follow.

Bulletin/ and scratch/

The reviewed IDC bulletin for the two weeks preceding the run time is located in `Bulletin/`. The name of this file is `TMbulletin.IDCREB`. Seismic events in this bulletin are plotted on the station availability map, as shown in Figure 4 on page 11.

The scripts *detplot*, *makecdf.2562*, *plotuptime*, and *tmstatus* write temporary files in the corresponding subdirectories in `scratch/` (see "scripts" on page 18).

```
<Session>/┬ Bulletin/— TMbulletin.IDCREB
          │
          ├ Region/┬ process.parameters
          │        ├ selected.phases
          │        ├ selected.stations
          │        ├ targets
          │        ├ tm_types
          │        │
          │        ├ Tm_types/–<tmtype>/──┬ <stn>/ ┬ selected.phases
          │        │                      │        └ <phase>/ — numeric.code
          │        │                      │
          │        │                      └ selected.stations
          │        │
          │        ├ STA/──── <stn>/ ─────┬ DFXbeams
          │        │                      ├ TMbeams
          │        │                      ├ Data/ ┬ cf.length
          │        │                      │       ├ data.interval
          │        │                      │       └ TMnnnn
          │        │                      │
          │        │                      └ Beams/— TMnnnn
          │        │
          │        └ TM/──────┬ TM_recipes/┬ Target_nnn
          │                   │            └ trvminmax
          │                   │
          │                   └ <tmtype>/──┬ RDF/──┬ data.interval
          │                                │       └ grid.rdf
          │                                │
          │                                └ Data/─┬ cf.length
          │                                        ├ data.interval
          │                                        └ Target_nnnn
          └ scratch/┬ detplot/
                    │
                    ├ makecdf/
                    │
                    ├ plotuptime/
                    │
                    └ tmstatus/
```

FIGURE 19. SESSION DIRECTORY STRUCTURE

Region/

Five ASCII files in `Region/` contain information pertinent to the whole processing session. The first file is `process.parameters`, which contains the parameters *magstdev*, *magconf*, *search1*, *search2*, *tmsamp*, *tmlen*, *stasamp*, and *stalen*, written in free format (see Table 20 on page 104 for a description of these parameters). The `targets` file is a copy of the appropriate target list described in "Targets/" on page 96. The remaining three files (in the following list) all have a two-column format with a name in the first column and an index number (line number) in the second column.

`selected.stations`	Lists the stations of interest (for example, `ABKT`, `ARCES`, and so on).
`selected.phases`	Lists the phases of interest (for example, `PTM`).
`tm_types`	Lists the tmtypes of interest (currently only `Network`).

Region/STA/

The `Region/STA/` directory includes a <*stn*>/ subdirectory for each station of interest (for example, `ARCES`), each having two subdirectories (`Beams/` and `Data/`). Each <*stn*>/ subdirectory also includes two ASCII files: `TMbeams` and `DFXbeams`. `DFXbeams` contains the list of STA beams written by the program *DFX*. `TMbeams` contains the list of TM beam definitions, which (for arrays) includes missteering parameters.

Names of beams (and corresponding files in <*stn*>`/Beams/` and <*stn*>`/Data/`) follow the format:

`TM`*nnn*

where *nnn* is a three-digit sequence number with leading zeroes if needed (for example, `001`).

The STA traces for each station are written in disk loops found in *<stn>*`/Data/`. The length of these files (in seconds) is found in *<stn>*`/Data/cf.length`. The starting and ending times (in format *YYYY-DDD:HH.MM.SS.sss*) are found in *<stn>*`/Data/data.interval`.

The ASCII processing recipes included in the files listed in `TMbeams` are found in *<stn>*`/Beams/` and contain the following parameters:

1. beam type (`B` for beam, `S` for single)
2. azimuth of beam (degrees)
3. slowness of beam (s/km)
4. STA file from *DFX* to be used for this beam (for example, `TM001`)
5. reference channel for beam from **site** table (for example, `ARCES`)
6. step size between centers of successive STA windows (seconds)
7. duration of STA window (seconds)
8. lower limit for distances covered by this beam (degrees)
9. higher limit for distances covered by this beam (degrees)
10. magnitude bias due to filtering and STA calculation (m_b units)
11. assumed signal loss (dB) at correct steering

A table of slowness values for signal loss assessment (s/km) and signal loss at those slowness values appears after these parameters.

Region/Tm_types/<tmtype>

`Region/Tm_types/` includes subdirectories for each tmtype (currently, the only tmtype is `Network`), each having a full complement of *<stn>*`/` subdirectories. Each of the *<stn>*`/` directories has one *<phase>*`/` subdirectory for each phase of interest (for example, `PTM`). The *<tmtype>*`/` directories contain a copy of the `selected.stations` file, and each *<tmtype>*`/`*<stn>*`/` directory contains a copy of `selected.phases`. Every *<tmtype>*`/`*<stn>*`/`*<phase>*`/` directory contains a `numeric.code` file, which contains a code calculated using the index numbers of the current station and phase (from `selected.stations` and `selected.phases`) as follows:

code = *stn_index* x 10 + *phase_index*

This value is always a three-digit number with a leading zero if necessary.

Region/TM/TM_recipes/

`TM_recipes/` contains an ASCII file for each target named `Target_`*nnnn*, where the sequence numbers (*nnnn*) begin with 1001. The format of the `Target_`*nnnn* files follows:

1. sequence number (for example, `1001`)
2. target name (for example, `Target_1001`)
3. latitude (degrees) of the target
4. longitude (degrees) of the target
5. depth (km, usually 0) of the target
6. radius of the target region (degrees)

Two lines follow for each STA trace to be used for that target. The first line includes the following parameters:

1. numeric code (from the `numeric.code` file, for example, `011` for station 1 and phase 1)
2. fully specified directory of STA trace (for example, <*Session*>`/Region/STA/`<*stn*>`/Data`)
3. name of the data file for the beam (for example, `TM001`)
4. station (for example, `NORES`)
5. phase (for example, `PTM`)
6. distance from region to center of station (degrees)

The second line contains the following parameters:

1. travel-time (s) between station and center of target region for phase (given above)
2. travel-time tolerance (s), lower limit
3. travel-time tolerance (s), higher limit
4. TMlen (length of TM window in seconds)
5. amplitude correction factor (added to log A/T to estimate magnitude)
6. assumed standard deviation of magnitude estimate

These two lines may be repeated if a target references several beams; the largest of the amplitudes from the beams is used for the target. After this information has been specified for all targets, the index number and *tmtype* (from `Region/tm_types/`) are found, followed by a list of the relevant numeric codes. The output directory and filename are listed at the bottom of the file.

An example target file for ARCES follows:

```
1001 Target_1001 90.0000 0.0000 0.0000 2.7400
011 /data/beet/Region/STA/ARCES/Data TM001 ARCESS PTM 20.5920
279 239.96 322.39 1.00 3.0183 0.3800
01 Network 011
```

Along with the 2,562 `Target_`*nnnn* files, `trvminmax` is an ASCII table of travel-time intervals for each station. The columns are the station name, the station index number (as it would appear in the file `selected.stations`), and the minimum and maximum travel-times for all targets around the world for this station. These times are adjusted for the radius of the target region (this adjustment can result in a negative number of seconds as it does in the ARCES case).

The corresponding `trvminmax` file for ARCES follows:

```
ARCES 01 -8.00 1340.15
```

Region/TM/<tmtype>/

`Region/TM/`*<tmtype>*`/RDF/` contains `grid.rdf`, a binary file that is a disk loop for storing the TM maps. The start- and end-times (*YYYY-DDD:HH.MM.SS.sss*) relevant to this file are listed in `data.interval` (in the same directory).

Detection thresholds for each target are stored in the disk loops named `Target_`*nnnn*. The lengths of these files are stored in `cf.length`, and the relevant start- and end-times are found in `data.interval` (with format *YYYY-DDD:HH.MM.SS.sss*). These two files are found in *<tmtype>*`/Data/`, along with the `Target_`*nnnn* files.

VALIDATING INSTALLATION

Proper operation of TM is confirmed by the presence of log files in `$LOGDIR/`*yyyyddd*`/TM/` and `$LOGDIR/`*yyyyddd*`/tuxshell/`. The log files should be checked for messages indicating problems with TM.

References

The following sources supplement or are referenced in the document:

[Han85] Hannon, W., "Seismic Verification of a Comprehensive Test Ban," *Science*, Volume 227, pp. 251–257, 1985.

[Har85] Harjes, H.P., "Global Seismic Network Assessment for Teleseismic Detection of Underground Nuclear Explosions," *Journal of Geophysics*, Volume 57, pp. 1–13, 1985.

[IDC5.1.1Rev2] Science Applications International Corporation, Veridian Pacific-Sierra Research, *Database Schema, Revision 2*, SAIC-00/3057, PSR-00/TN2830, 2000.

[IDC5.1.3Rev0.1] Science Applications International Corporation, Veridian Pacific-Sierra Research, *Configuration of PIDC Databases,* SAIC-01/3022, 2001.

[IDC6.2.1] Science Applications International Corporation, *Release 2 – Operation and Maintenance – Seismic, Hydroacoustic, and Infrasonic System,* SAIC-00/3000, 2000.

[IDC6.5.2Rev0.1] Science Applications International Corporation, *Distributed Application Control System (DACS) Software User Manual,* SAIC-00/3038, 2000.

[IDC7.1.14] Science Applications International Corporation, *Threshold Monitoring Subsystem,* SAIC-01/3015, 2001.

[Ken91b] Kennett, B., and Engdahl, E., "Traveltimes for Global Earthquake Location and Phase Identification," *Geophysical Journal International*, Volume 105, pp. 429–465, 1991.

[Kvæ92] Kværna, T., "Initial Results From Global Generalized Beamforming," *Semiannual Technical Summary, 1 April – 30 September 1992,* NORSAR Scientific Report 1-92/93, NORSAR, Kjeller, Norway, 1992.

[Kvæ94] Kværna, T., Ringdal, F., Iversen, H., and Larsen, N. H. K., "A System for Continuous Seismic Threshold Monitoring," *Final Report, Semiannual Technical Summary, 1 April – 30 September 1994,* NORSAR Scientific Report 1-94/95, NORSAR, Kjeller, Norway, 1994.

[Kvæ96] Kværna, T., "Tuning of Processing Parameters for Global Threshold Monitoring at the IDC," *Semiannual Technical Summary, 1 April – 30 September 1996*, NORSAR Scientific Report 1-96/97, NORSAR, Kjeller, Norway, 1996.

[Kvæ97a] Kværna, T., "Threshold Magnitudes," *Semiannual Technical Summary, 1 October 1996 – 31 March 1997,* NORSAR Scientific Report, 2-96/97, NORSAR, Kjeller, Norway, 1997.

[Kvæ97b] Kværna, T., Ringdal, F., Bassdshaug, U., and Iversen, H., "Status of the Global Threshold Monitoring (TM) System," *Semiannual Technical Summary, 1 April – 30 September 1997*, NORSAR Scientific Report, 1-97/98, NORSAR, Kjeller, Norway, 1997.

[Lil86] Lilwall, R. C. and Marshall, P. D., "Body Wave Magnitudes and Locations of Soviet Underground Explosions at the Novaya Zemlya Test Site," *AWRE Report Number 017/86*, Blacknest, U.K., 1986.

[Rin79] Ringdal, F. and Fyen, J., "Analysis of Global P-wave Attenuation Characteristics Using ISC Data Files," *Semiannual Technical Summary, 1 April – 30 September 1979,* NORSAR Scientific Report 1-79/80, NORSAR, Kjeller, Norway, 1979.

[Rin86] Ringdal, F., "Study of Magnitudes, Seismicity and Earthquake Detectability Using a Global Network," *Bulletin of the Seismological Society of America*, Volume 76, pp. 1641–1659, 1986.

[Rin89]	Ringdal, F., and Kværna, T., "A Multi-channel Processing Approach to Real-time Network Detection, Phase Association, and Threshold Monitoring," *Bulletin of the Seismological Society of America*, Volume 79, pp. 780–798, 1989.
[Rin92]	Ringdal, F. and Kværna, T., "Continuous Seismic Threshold Monitoring," *Geophysical Journal International*, Volume 111, pp. 505–514, 1992.
[Rin95]	Ringdal, F., Kværna, T., and Mykkeltveit, S., "Global Seismic Threshold Monitoring and Automated Network Processing," *Semiannual Technical Summary, 1 October 1994 – 31 March 1995*, NORSAR Scientific Report 2-94/95, NORSAR, Kjeller, Norway, 1995.
[Ser89]	Sereno, T. J. and Bratt, S. R., "Seismic Detection Capability at NORESS and Implications for the Detection Threshold of a Hypothetical Network in the Soviet Union," *Journal of Geophysical Research,* Volume 94, pp. 10397–10414, 1989.
[Syk82]	Sykes, L. and Evernden, J., "The Verification of a Comprehensive Nuclear Test Ban," *Scientific American*, Volume 247, pp. 47–55, 1982.
[Vei72]	Veith, K. F., and Clawson, G. E., "Magnitude of Short-period P Wave Data," *Bulletin of the Seismological Society of America*, Volume 62, pp. 435–453, 1972.
[Vin92]	Vinje, V., Iversen, E., Gjøystdal, H., and Åstebøl, K., "Traveltime and Amplitude Estimation Using Wavefront Construction," *Abstract of paper presented at the 54th Meeting and Technical Exhibition of the European Association of Exploration Geophysicists* in Paris, France, 1–5 June 1992.
[Wah96]	Wahl, D. D., *User's Manual for the Detection and Feature Extraction Program*, Science Applications International Corporation, SAIC-96/1098, 1996.
[Wes95]	Wessel, P. and Smith, H. F., "New version of Generic Mapping Tools Released," *EOS Transactions, American Geophysical Union*, Volume 76, pp. 329, 1995.

[WGB98a] Working Group B, *Initial Draft of the Operational Manual for the International Data Centre.* Preparatory Commission of the Comprehensive Nuclear-Test-Ban Organization, CTBT/PC/V/WGB/TL/44/Rev.2, 1998.

Appendix: PIDC Tuning Results

This appendix documents the station tuning results for the arrays and 3-C stations in use at the PIDC at the time of publication (Tables A-1 through A-4).

Table A-1: Array Configurations

Array	Configuration	sta_chan
ARCES	a0-b-c-d	ARA0_sz ARB1_sz ARB2_sz ARB3_sz ARB4_sz ARB5_sz ARC1_sz ARC2_sz ARC3_sz ARC4_sz ARC5_sz ARC6_sz ARC7_sz ARD1_sz ARD2_sz ARD3_sz ARD4_sz ARD5_sz ARD6_sz ARD7_sz ARD8_sz ARD9_sz
ASAR	vertical	AS01_sz AS02_sz AS03_sz AS04_sz AS05_sz AS06_sz AS07_sz AS08_sz AS09_sz AS10_sz AS11_sz AS12_sz AS13_sz AS14_sz AS15_sz AS16_sz AS17_sz AS18_sz AS19_sz
BRAR	vertical	BR01_sz BR02_sz BR03_sz BR04_sz BR05_sz BR06_sz
CMAR	vertical	CM01_sz CM02_sz CM03_sz CM04_sz CM05_sz CM06_sz CM07_sz CM10_sz CM12_sz CM13_sz CM14_sz
ESDC	vertical	ES01_sz ES02_sz ES03_sz ES04_sz ES05_sz ES06_sz ES07_sz ES08_sz ES09_sz ES10_sz ES11_sz ES12_sz ES13_sz ES14_sz ES15_sz ES16_sz ES17_sz ES18_sz ES19_sz ESLA_sz
FINES	vertical	FIA0_sz FIA1_sz FIA2_sz FIA3_sz FIB1_sz FIB2_sz FIB3_sz FIB4_sz FIB5_sz FIB6_sz FIC1_sz FIC2_sz FIC3_sz FIC4_sz FIC5_sz FIC6_sz
GERES	group6	GEA1_sz GEA2_sz GEB1_sz GEB2_sz GEB4_sz GEB5_sz GEC1_sz GEC2_sz GEC3_sz GEC4_sz GEC5_sz GEC6_sz GEC7_sz GED1_sz GED2_sz GED3_sz GED4_sz GED5_sz GED6_sz GED7_sz GED8_sz GED7_sz

TABLE A-1: ARRAY CONFIGURATIONS (CONTINUED)

Array	Configuration	sta_chan
ILAR	vertical	IL01_sz IL02_sz IL03_sz IL04_sz IL05_sz IL06_sz IL07_sz IL08_sz IL09_sz IL10_sz IL11_sz IL12_sz IL13_sz IL14_sz IL15_sz IL16_sz IL17_sz IL18_sz IL19_sz
KSAR	vertical	KS01_sz KS02_sz KS03_sz KS04_sz KS05_sz KS06_sz KS07_sz KS08_sz KS09_sz KS10_sz KS11_sz KS12_sz KS13_sz KS14_sz KS15_sz KS16_sz KS17_sz KS18_sz KS19_sz
MJAR	vertical	MJ00_ez MJ01_ez MJ02_ez MJ03_ez MJ04_ez MJ05_ez
NOA	NB2	NB201_sz NB202_sz NB203_sz NB204_sz NB205_sz NB200_sz
NVAR	vertical	NV01_sz NV02_sz NV03_sz NV04_sz NV05_sz NV06_sz NV07_sz NV08_sz NV09_sz NV10_sz
PDAR	vertical	PD01_sz PD02_sz PD03_sz PD04_sz PD05_sz PD06_sz PD07_sz PD08_sz PD09_sz PD10_sz PD11_sz PD12_sz PD13_sz
PDYAR	vertical	PDY01_sz PDY02_sz PDY03_sz PDY04_sz PDY05_sz PDY06_sz PDY07_sz PDY08_sz PDY09_sz
TXAR	vertical	TX01_sz TX02_sz TX03_sz TX04_sz TX06_sz TX07_sz TX08_sz TX09_sz TX10_sz
WRA	grouptm	WR1_sz WR2_sz WR3_sz WR4_sz WR5_sz WB1_sz WB2_sz WB3_sz WB4_sz WB5_sz
YKA	grouptm	YKB3_sz YKB4_sz YKB6_sz YKB7_sz YKR5_sz YKR6_sz YKR7_sz YKR8_sz YKR9_sz

TABLE A-2: ARRAY TUNING PARAMETERS

Array Configuration	Distance Interval (deg.)	Frequency Band (Hz)	A/T Correction (m_b units)	Signal Loss (dB)	3 dB Level (s/km)	Beams
ARCES a0-b-c-d	0–15	2.5 – 8.0	–0.072±0.200	1.370	0.043	9
	15–180	1.5 – 6.0	0.209±0.176	0.541	0.106	4
ASAR vertical	0–180	1.0 – 4.5	0.09±0.069	0.50	0.033	19
BRAR	0–180	0.8 – 3.0	–0.011±0.057	0.425	0.148	5
CMAR vertical	0–15	0.8 – 3.0	0.01±0.054	0.62	0.036	11
	15–112	0.8 – 3.0	0.01±0.054	0.31	0.044	12
	112–180	0.8 – 3.0	0.01±0.054	0.52	0.052	1
ESDC vertical	0–180	1.0 – 4.5	0.09±0.058	1.04	0.034	19
FINES vertical	0–15	2.5 – 8.0	0.00±0.512	1.23	0.057	7
	15–180	1.5 – 6.0	0.09±0.108	0.18	0.131	1
GERES group6	0–20	1.25–4.5	0.05±0.120	0.71	0.073	6
	20–180	0.8–3.0	0.07±0.170	0.12	0.116	1
ILAR vertical	0–15	1.25–4.5	0.16±0.054	1.80	0.025	16
	15–180	1.0–4.5	0.09±0.059	1.27	0.038	19
KSAR vertical	0–20	0.8–3.0	0.05±0.166	0.90	0.034	11
	20–180	0.8–3.0	0.02±0.063	0.42	0.043	10
MJAR vertical	0–20	1.25–4.5	0.43±0.059	2.96	0.017	26
	20–180	0.8–3.0	0.37±0.032	1.48	0.025	24
NOA NB2	0–180	1.0–4.5	0.03±0.117	1.31	0.036	19
NVAR vertical	0–180	0.8–3.0	–0.057±0.140	1.050	0.086	6
PDAR vertical	0–10	2.0–8.0	0.20±0.043	4.79	0.060	7
	10–180	0.8–3.0	0.01±0.036	0.18	0.116	8
PDYAR vertical	0–180	0.8–4.5	–0.049±0.048	1.280	0.096	6
TXAR	0–180	0.8–4.5	0.18±0.097	0.20	0.084	7

TABLE A-2: ARRAY TUNING PARAMETERS (CONTINUED)

Array Configuration	Distance Interval (deg.)	Frequency Band (Hz)	A/T Correction (m_b units)	Signal Loss (dB)	3 dB Level (s/km)	Beams
WRA grouptm	0–20	1.5–6.0	0.05±0.069	2.87	0.027	15
	20–180	1.25–4.5	0.08±0.078	1.34	0.028	24
YKA grouptm	0–180	0.8–3.0	0.05±0.079	1.46	0.032	19

TABLE A-3: ARRAY BEAM STEERING PARAMETERS

Array	Distance Interval (deg.)	Azimuth (deg.) and Slowness (s/km) of Beams					
ARCES	0–15	0.0, 0.111	40.0, 0.111	80.0, 0.111	115.0, 0.111	160.0, 0.111	150.0, 0.111
		240.0, 0.111	280.0, 0.111	315.0, 0.111			
ARCES	15–180	0.0, 0.059	90.0, 0.059	180.0, 0.059	270.0, 0.059		
ASAR	0–180	206.6, 0.118	180.0, 0.105	153.4, 0.118	236.3, 0.095	206.6, 0.059	153.4, 0.059
		123.7, 0.095	270.0, 0.105	270.0, 0.053	0.0, 0.000	90.0, 0.053	90.0, 0.105
		303.7, 0.095	333.4, 0.059	26.6, 0.059	56.3, 0.095	333.4, 0.118	0.0, 0.105
		26.6, 0.118					
BRAR	0–180	0.0, 0.000	0.0, 0.123	90.0, 0.123	180.0, 0.123	270.0, 0.123	

TABLE A-3: ARRAY BEAM STEERING PARAMETERS (CONTINUED)

Array	Distance Interval (deg.)	Azimuth (deg.) and Slowness (s/km) of Beams					
CMAR	0–15	0.0, 0.123	32.7, 0.123	65.5, 0.123	98.2, 0.123	130.9, 0.123	163.6, 0.123
		196.4, 0.123	229.1, 0.12	261.8, 0.123	294.5, 0.123	327.3, 0.123	
	15–112	0.0, 0.000	0.0, 0.086	32.7, 0.086	65.5, 0.086	98.2, 0.086	130.9, 0.086
		163.6, 0.086	196.4, 0.086	229.1, 0.086	261.8, 0.086	294.5, 0.086	327.3, 0.086
	112–180	0.0, 0.000					
ESDC	0–180	206.6, 0.121	180.0, 0.108	153.4, 0.121	236.3, 0.098	206.6, 0.060	153.4, 0.060
		123.7, 0.098	270.0, 0.108	270.0, 0.054	0.0, 0.000	90.0, 0.054	90.0, 0.108
		303.7, 0.098	333.4, 0.060	26.6, 0.060	56.3, 0.098	333.4, 0.121	0.0, 0.108
		26.6, 0.121					
FINES	0–15	0.0, 0.123	51.4, 0.123	102.9, 0.123	154.3, 0.123	205.7, 0.123	257.1, 0.123
		308.6, 0.123					
	15–180	0.0, 0.000					
GERES	0–20	0.0, 0.111	60.0, 0.111	120.0, 0.111	180.0, 0.111	240.0, 0.111	300.0, 0.111
	20–180	0.0, 0.000					

TABLE A-3: ARRAY BEAM STEERING PARAMETERS (CONTINUED)

Array	Distance Interval (deg.)	Azimuth (deg.) and Slowness (s/km) of Beams					
ILAR	0–15	0.0, 0.123	22.5, 0.123	45.0, 0.123	67.5, 0.123	90.0, 0.123	112.5, 0.123
		135.0, 0.123	157.5, 0.123	180.0, 0.123	202.5, 0.123	225.0, 0.123	247.5, 0.123
		270.0, 0.123	292.5, 0.123	315.0, 0.123	337.5, 0.123		
	15–180	206.6, 0.136	180.0, 0.122	153.4, 0.136	236.3, 0.110	206.6, 0.068	153.4, 0.068
		23.7, 0.110	270.0, 0.122	270.0, 0.061	0.0, 0.000	90.0, 0.061	90.0, 0.122
		303.7, 0.110	333.4, 0.068	26.6, 0.068	56.3, 0.110	333.4, 0.136	0.0, 0.122
		26.6, 0.136					
KSAR	0–20	0.0, 0.111	32.7, 0.111	65.5, 0.111	98.2, 0.111	130.9, 0.111	163.6, 0.111
		196.4, 0.111	229.1, 0.111	261.8, 0.111	294.5, 0.111	327.3, 0.111	
	20–180	225.0, 0.097	180.0, 0.068	135.0, 0.097	270.0, 0.102	270.0, 0.034	90.0, 0.034
		90.0, 0.102	315.0, 0.097	0.0, 0.068	45.0, 0.097		

TABLE A-3: ARRAY BEAM STEERING PARAMETERS (CONTINUED)

Array	Distance Interval (deg.)	Azimuth (deg.) and Slowness (s/km) of Beams					
MJAR	0–20	0.0, 0.111	13.8, 0.111	27.7, 0.111	41.5, 0.111	55.4, 0.111	69.2, 0.111
		83.1, 0.111	96.9, 0.111	110.8, 0.111	124.6, 0.111	138.5, 0.111	152.3, 0.111
		166.2, 0.111	180.0, 0.111	193.8, 0.111	207.7, 0.111	221.5, 0.111	235.4, 0.111
		249.2, 0.111	263.1, 0.111	276.9, 0.111	290.8, 0.111	304.6, 0.111	318.5, 0.111
		332.3, 0.111	346.2, 0.111				
	20–180	216.9, 0.101	194.0, 0.083	166.0, 0.083	143.1, 0.101	243.4, 0.090	225.0, 0.057
		180.0, 0.040	135.0, 0.057	116.6, 0.090	270.0, 0.101	270.0, 0.060	270.0, 0.020
		90.0, 0.020	90.0, 0.060	90.0, 0.101	296.6, 0.090	315.0, 0.057	0.0, 0.040
		45.0, 0.057	63.4, 0.090	323.1, 0.101	346.0, 0.083	14.0, 0.083	36.9, 0.101
NOA	0–180	206.6, 0.126	180.0, 0.113	153.4, 0.126	236.3, 0.102	206.6, 0.063	153.4, 0.063
		123.7, 0.102	270.0, 0.113	270.0, 0.056	0.0, 0.000	90.0, 0.056	90.0, 0.113
		303.7, 0.102	333.4, 0.063	26.6, 0.063	56.3, 0.102	333.4, 0.126	0.0, 0.113
		26.6, 0.126					
NVAR	0–180	0.0, 0.000	0.0, 0.123	72.0, 0.123	144.0, 0.123	216.0, 0.123	288.0, 0.123

TABLE A-3: ARRAY BEAM STEERING PARAMETERS (CONTINUED)

Array	Distance Interval (deg.)	Azimuth (deg.) and Slowness (s/km) of Beams					
PDAR	0–10	0.0, 0.123	51.4, 0.123	102.9, 0.123	154.3, 0.123	205.7, 0.123	257.1, 0.123
		308.6, 0.123					
	10–180	0.0, 0.000	0.0, 0.116	51.4, 0.116	102.9, 0.116	154.3, 0.116	205.7, 0.116
		257.1, 0.116	308.6, 0.116				
PDYAR	0–180	0.0, 0.000	0.0, 0.123	72.0, 0.123	144.0, 0.123	216.0, 0.123	288.0, 0.123
TXAR	0–180	206.6, 0.116	153.4, 0.116	270.0, 0.104	0.0, 0.000	90.0, 0.104	333.4, 0.116
		26.6, 0.116					
WRA	0–20	0.0, 0.111	24.0, 0.111	48.0, 0.111	72.0, 0.111	96.0, 0.111	120.0, 0.111
		144.0, 0.111	168.0, 0.111	192.0, 0.111	216.0, 0.111	240.0, 0.111	264.0, 0.111
		288.0, 0.111	312.0, 0.111	336.0, 0.111			
	20–180	216.9, 0.109	194.0, 0.090	166.0, 0.090	143.1, 0.109	243.4, 0.098	225.0, 0.062
		180.0, 0.044	135.0, 0.062	116.6, 0.098	270.0, 0.109	270.0, 0.066	270.0, 0.022
		90.0, 0.022	90.0, 0.066	90.0, 0.109	296.6, 0.098	315.0, 0.062	0.0, 0.044
		45.0, 0.062	63.4, 0.098	323.1, 0.109	346.0, 0.090	14.0, 0.090	36.9, 0.109

Table A-3: Array Beam Steering Parameters (continued)

Array	Distance Interval (deg.)	Azimuth (deg.) and Slowness (s/km) of Beams					
YKA	0–180	206.6, 0.121	180.0, 0.109	153.4, 0.121	236.3, 0.098	206.6, 0.061	153.4, 0.061
		123.7, 0.098	270.0, 0.109	270.0, 0.054	0.0, 0.000	90.0, 0.054	90.0, 0.109
		303.7, 0.098	333.4, 0.061	26.6, 0.061	56.3, 0.098	333.4, 0.121	0.0, 0.109
		26.6, 0.121					

Table A-4: 3-C Station Tuning Parameters

Station	Distance Interval (deg.)	Sensor	Frequency Band (Hz)	A/T Correction (m_b units)
ABKT	0–180	ABKTbz	0.8–3.0	0.03±0.076
BDFB	0–180	BDFB_bz	1.0–4.5	0.11±0.055
BGCA	0–180	BGCA_bz	1.25–4.5	0.19±0.076
BJT	0–180	BJT_bz	0.8–3.0	–1.88±0.150
BOSA	0–180	BOSA_bz	1.25–4.5	0.23±0.117
CPUP	0–20	CPUP_bz	1.25–4.5	0.15±0.062
CPUP	20–180	CPUP_bz	1.0–4.5	0.18±0.100
DBIC	0–180	DBIC_bz	1.25–4.5	0.19±0.080
HIA	0–180	HIA_bz	0.8–3.0	0.03±0.051
KBZ	0–180	KBZ_sz	0.8–4.5	0.12±0.183
LPAZ	0–20	LPAZ_bz	1.25–4.5	0.17±0.063
LPAZ	20–180	LPAZ_bz	1.0–4.5	0.12±0.070
MAW	0–180	MAW_bz	1.0–4.5	0.11±0.160

TABLE A-4: 3-C STATION TUNING PARAMETERS (CONTINUED)

Station	Distance Interval (deg.)	Sensor	Frequency Band (Hz)	A/T Correction (m_b units)
MNV	0–180	MNV_bz	0.8–3.0	0.01±0.047
NRI	0–180	NRI_sz	0.8–4.5	0.25±0.160
PLCA	0–20	PLCA_bz	1.5–6.0	0.18±0.057
PLCA	20–180	PLCA_bz	1.25–4.5	0.26±0.107
ROSC	0–180	ROSC_bz	0.8–3.0	0.06±0.109
SCHQ	0–15	SCHQ_bz	2.0–8.0	–1.02±0.121
SCHQ	15–20	SCHQ_bz	1.5–6.0	–0.77±0.121
STKA	0–180	STKA_bz	1.5–6.0	0.29±0.105
ULM	0–180	ULM_bz	1.0–4.5	–0.88±0.047
VNDA	0–180	VNDA_bz	1.25–4.5	0.26±0.121
ZAL	0–180	ZAL_sz	0.8–4.5	0.30±0.160

Glossary

Symbols

3-C

Three-component.

A

amplitude

Zero-to-peak height of a waveform in nanometers.

array

Collection of sensors distributed over a finite area (usually in a cross or concentric pattern) and referred to as a single station.

arrival

Signal that has been associated to an event. First, the Global Association (*GA*) software associates the signal to an event. Later during interactive processing, many arrivals are confirmed and improved by visual inspection.

ASCII

American Standard Code for Information Interchange. Standard, unformatted 256-character set of letters and numbers.

AutoDRM

Automatic Data Request Manager.

azimuth

Direction, in degrees clockwise with respect to North, from a station to an event.

B

back azimuth

Direction, in degrees, from an event to the station.

background noise

Natural movements of the earth, oceans, and atmosphere as seen on S/H/I sensors (usually measured in data preceding signals).

beam

(1) Waveform created from array station elements that are sequentially summed in the direction of a specified azimuth and slowness. (2) Any derived waveform (for example, a filtered waveform).

beam recipe

Table of parameters that describe how to create and use a beam. Parameters may include the channels to use in forming the beam, the filter to use, the direction and slowness in which to steer the beam, and the snr threshold to apply for detection.

beamform

Sum a set of waveforms from array station elements with time delays introduced to compensate for the time it takes a wave to travel across the array.

branch

Discrete ray paths through the outer and inner core, for example, in the PKPbc branch of the PKP phase.

bulletin

Chronological listing of event origins spanning an interval of time. Often, the specification of each origin or event is accompanied by the event's arrivals and sometimes with the event's waveforms.

C

CD-ROM

Compact Disk–Read Only Memory.

channel

Component of motion or distinct stream of data.

child process

UNIX process created by the *fork* routine. The child process is a snapshot of the parent at the time it called *fork*.

CDF

Common Data Format.

coherent

Quality of having a fixed phase relationship; as signals from a wavefront detected on numerous seismic or infrasonic array station elements.

coherent beam

Summation of data from numerous seismic or infrasonic array station elements after shifting the data traces in time to maximize the coherence of plane-wave signals travelling along a particular azimuth and slowness.

component

(1) One dimension of a three-dimensional signal; (2) The vertically or horizontally oriented (north or east) sensor of a station used to measure the dimension; (3) One of the parts of a system; also referred to as a module or unit.

Computer Software Component

Functionally or logically distinct part of a computer software configuration item, typically an aggregate of two or more software units.

Computer Software Configuration Item

Aggregation of software that is designated for configuration management and treated as a single entity in the configuration management process.

configuration

(1) (hardware) Arrangement of computer system or component as defined by the number, nature, and interconnec-

tion of its parts. (2) (software) Set of adjustable parameters, usually stored in files, for applications to use at run time.

continuous waveform data

Waveform data that are transmitted to the IDC on a nominally continuous basis.

COTS

Commercial-Off-the-Shelf; terminology that designates products such as hardware or software that can be acquired from existing inventory and used without modification.

CSC

Computer Software Component.

CSCI

Computer Software Configuration Item.

CTBT

Comprehensive Nuclear-Test-Ban Treaty (the Treaty).

D

D

Days.

DACS

Distributed Application Control System. This software supports inter-application message passing and process management.

data flow

Sequence in which data are transferred, used, and transformed during the execution of a computer program.

dB

Decibel.

DFX

Detection and Feature Extraction. *DFX* is a programming environment that executes applications written in Scheme (known as *DFX* applications).

disk loop

Storage device that continuously stores new waveform data while simultaneously deleting the oldest data on the device.

E

epoch time

Number of seconds after 1 January, 1970 00:00:00.0.

event

Unique source of seismic, hydroacoustic, or infrasonic wave energy that is limited in both time and space.

F

filesystem

Named structure containing files in subdirectories. For example, UNIX can support many filesystems; each has a unique name and can be attached (or mounted) anywhere in the existing file structure.

fork

UNIX system routine that is used by a parent process to create a child process.

FTP

File Transfer Protocol; protocol for transferring files between computers.

G

GB

Gigabyte. A measure of computer memory or disk space that is equal to 1,024 megabytes.

H

H

Hour(s).

Hz

Hertz.

I

I/O

Input/Output.

IASPEI

International Association of Seismology and Physics of the Earth's Interior.

IDC

International Data Centre.

IMS

International Monitoring System.

instance

Running computer program. An individual program may have multiple instances on one or more host computers.

IPC

Interprocess Communication. The messaging system by which applications communicate with each other through libipc common library functions. See *tuxshell*.

K

km

Kilometer.

L

logon

Process of a computer user identifying themselves by means of a user ID and password to a computer operating system as a registered user, beginning an interactive session.

LTA (or LTAV)

Long Term Average. A running average of the absolute value or squared value of a waveform. The averaging window is long compared to the short-term averaging window.

M

magnitude

Empirical measure of the size of an event (usually made on a log scale).

mask

Array of start and end indices for defective data samples in a time series used to remove the defective data from a time series.

m_b

Magnitude based on seismic body waves.

N

network

Spatially distributed collection of seismic, hydroacoustic, or infrasonic stations for which the station spacing is much larger than a wavelength.

noise

Incoherent natural or artificial perturbations of the waveform trace caused by ice, animals migrations, cultural activity, equipment malfunctions or interruption of satellite communication, or ambient background movements.

O

origin

Hypothesized time and location of a seismic, hydroacoustic, or infrasonic event. Any event may have many origins. Characteristics such as magnitudes and error estimates may be associated with an origin.

origin beam

Coherent beam steered to the estimated event origin.

P

P Phase

Seismic wave that travels from the event to the station as a compressional wave through the solid earth.

par

See parameter.

parameter

User-specified token that controls some aspect of an application (for example, database name, threshold value). Most parameters are specified using [*token* = *value*] strings, for example, `dbname=mydata/base@oracle`.

parameter (par) file

ASCII file containing values for parameters of a program. Par files are used to replace command line arguments. The files are formatted as a list of `[`*token* `=` *value*`]` strings.

parent process

UNIX process that creates a child process by the *fork* routine. The child process is a snapshot of the parent at the time it called *fork*.

pathname

Filesystem specification for a file's location.

phase

Arrival that is identified based on its path through the earth.

PIDC

Prototype International Data Centre.

pipeline

Flow of data at the IDC from the receipt of communications to the final automated processed data before analyst review.

PKP Phase

Seismic wave that travels from the event to the station as a compressional wave through the crust, mantle, and outer core.

primary seismic

IMS seismic station(s) or data that is (are) part of the detection network.

primary stations

Stations that make up the primary seismic network of the IMS. Primary stations send data continuously to the IDC.

probability of detection

Probability estimate that an arrival from a given event will be detected at a station given the location and magnitude of the event, the average noise level and its standard deviation at the station, and the signal-to-noise detection threshold.

product

Bulletins, data, and other information collected, produced, and distributed by the IDC.

R

RDF

Regional Data Format.

real time

Actual time during which something takes place.

REB

Reviewed Event Bulletin; the bulletin formed of all S/H/I events that have passed analyst inspection and quality assurance review.

reference channel

Array element to which the station's timing is referenced with respect to its other elements, reflecting the timing of the array as a whole. This channel is typically either the element at the center of a circular array or the element at the intersection of a cross-shaped array.

regional

(1) (distance) Source to seismometer separations between a few degrees and 20 degrees. (2) (event) Recorded at distances where the first P and S waves from shallow events have traveled along paths through the uppermost mantle.

run

(1) Single, usually continuous, execution of a computer program. (2) To execute a computer program.

S

s

Second(s) (time).

S

Second(s) (time).

SAIC

Science Applications International Corporation.

sbsnr

Signal-to-noise ratio measured on a standard beam.

script

Small executable program, written with UNIX and other related commands, that does not need to be compiled.

seismic

Pertaining to elastic waves traveling through the earth.

SEL3

Standard Event List 3; S/H/I bulletin created by totally automatic analysis of both continuous data and segments of data specifically down-loaded from stations of the auxiliary seismic network. Typically, the list runs 12 hours behind real time.

slowness

Inverse of velocity, in seconds/degree; a large slowness has a low velocity.

SMR

Software Modification Request.

snr

Signal-to-noise ratio.

spectrum (spectral)

Plot of the energy contained in waveforms as a function of frequency.

STA (or STAV)

Short-term average. A running average of the absolute value or squared value of a waveform. The averaging window is short in duration compared to the LTA.

standard beam

Beam in which the delays between stations are based on a pre-defined set of recipe parameters (referred to as the standard recipe parameters).

station

Collection of one or more monitoring instruments. Stations can have either one sensor location (for example, BGCA) or a spatially distributed array of sensors (for example, ASAR).

station processing

Processing based on data from a single station.

T

tar

Tape archive. UNIX command for storing or retrieving files and directories. Also used to describe the file or tape that contains the archived information.

teleseismic

(1) (distance) Source to seismometer separations of 20 degrees or more. (2) (event) Recorded at distances where the first P and S waves from shallow events have traveled paths through the mantle/core.

TM

Threshold monitoring. A technique to keep track of the minimum detectable event based on noise levels at stations.

Treaty

Comprehensive Nuclear-Test-Ban Treaty (CTBT).

tuxshell

Process in the Distributed Processing CSCI used to execute and manage applications. See IPC.

U

UNIX

Trade name of the operating system used by the Sun workstations.

W

waveform

Time-domain signal data from a sensor (the voltage output) where the voltage has been converted to a digital count (which is monotonic with the amplitude of the stimulus to which the sensor responds).

Y

Y

Year.

Index

Numerics

3-C station tuning parameters A9

A

adding stations 29
AddTMStation 14, 15, 16, 17, 23, 29, 31, 63, 64, 65, 68, 70
    error messages 63
array
    beam steering parameters A4
    configurations A1
    tuning parameters A3
AutoDRM 37

B

`beambasic` parameters 88
beam deployment 46
beamforming
    signal loss 43
`Beamset/` 88
`Bulletin/` 107

C

CopyPSFile 14, 15, 18
CreateTMSession 8, 14, 15, 16, 17, 19, 23, 26, 29, 31, 33, 50, 51, 56, 57, 58, 59, 60, 63, 64, 65, 66, 68, 70, 85, 99, 103, 106
    error messages 63
    input 104
    output 106
`cur_pri.data` parameters 95

D

DACS parameters 19
database variables 79
`data.cpt` parameter files 94
data flow diagram 22
data flow symbols iv
DeleteTMStation 14, 15, 16, 17, 23, 30, 33, 63, 64, 68, 70
    error messages 63
deleting stations 32
detplot 6, 8, 18, 107
DFX 2, 15, 19, 21, 26, 29, 30, 31, 32, 64, 79
    beam configuration 81
    beam weight parameters 84
    detection parameters 79
    recipe parameters 81
    TM beam configuration 81
    TM parameters 80
disk loop
    copying 33

E

error messages
    AddTMStation 63
    CreateTMSession 63
    DeleteTMStation 63

general 52
TMmap 60
TMthreshold 55

F

fileproduct 98
fileproducts 28
fonts and color tables 20
fpdescription 98

G

global TM variables 77
GMT 96

I

interval 98
invoke, definition of v

L

LEBDB v
`libdata/` 90
log files 112
LoopCopy 14, 15, 16, 17, 23, 33, 34
loopuptime 6, 9, 18

M

makecdf.2562 6, 18, 107
makelibs 15, 16, 17, 76
makemods 15, 16, 17, 76
`missteer.`*Dlow-Dhig* parameters 89
mis-steering
signal loss 43
monitoring 50

O

`offset.data` parameters 96

P

`par/` 93
parameters 18
parameter tuning 36
performance characteristics 14
plotuptime 6, 8, 9, 18, 107
problem reporting 73
processing flow 4
processing hierarchy 7

R

rdf2cdf 6, 16
rebdone 5, 21, 26
`Region/` 108
`Region/STA/` 108
`Region/TM/TM_recipes/` 110
`Region/TM/`*tmtype*`/` 112
`Region/Tm_types/`*tmtype* 109
replotuptime 6, 9, 18, 30, 32

S

`scratch/` 107
Session directory structure 107
shared variables 78
shutdown 28
software environment 21
startup 26
static data 19
static data directory 87

T

`tab.PTM` parameters 92
`targets.`*n* parameters 97
tm_beambasic 14, 16, 50, 52, 65
tm_globrec 14, 15, 17, 50, 52, 68, 69, 70, 71, 72
tm_stast 6, 17, 20
TMbulletin 6, 9, 18, 30, 32
tmcfwrite 51
TMmap 4, 5, 6, 8, 16, 17, 19, 27, 30, 32, 33, 50, 52, 55, 60, 61, 62, 86, 99
    error messages 60
    parameters 87
TMpostreb 5, 6, 9, 18, 19, 26, 28, 32, 33, 50, 63, 79, 99
TMprod 5, 6, 8, 9, 18, 19, 27, 30, 31, 32, 33, 50, 63, 79, 93, 99
TM session
    adding stations 29
    creating 29
    deleting stations 32
`TM.site` parameters 90
tmstatus 6, 8, 18, 107
`tmstatus.par` parameters 93
TMthreshold 2, 4, 5, 8, 16, 17, 19, 26, 30, 32, 33, 50, 51, 52, 55, 56, 57, 58, 59, 60, 68, 69, 85, 99
    error messages 55
    parameters 85
tuning 36
Tuxedo files 99
tuxshell 50, 63, 99
    parameters 99
typographical conventions v

V

variables
    database 79
    global 77
    shared 78